On Cross-Layer Design for Streaming Video Delivery in Multiuser Wireless Environments

Lai-U Choi,[1] Wolfgang Kellerer,[2] and Eckehard Steinbach[1]

[1] *Media Technology Group, Institute of Communication Networks, Department of Electrical Engineering and Information Technology, Munich University of Technology, 80290 Munich, Germany*
[2] *Future Networking Lab, DoCoMo Communications Laboratories Europe GmbH, 80687 Munich, Germany*

We exploit the interlayer coupling of a cross-layer design concept for streaming video delivery in a multiuser wireless environment. We propose a cross-layer optimization between application layer, data link layer, and physical layer. Our aim is to optimize the end-to-end quality of the wireless streaming video application as well as efficiently utilizing the wireless resources. A possible architecture for achieving this goal is proposed and formulated. This architecture consists of the process of parameter abstraction, a cross-layer optimizer, and the process of decision distribution. In addition, numerical results obtained with different operating modes are provided. The results demonstrate the potential of this proposed joint optimization.

1. INTRODUCTION

Since the introduction of digital personal wireless networks around 1990, wireless communication has evolved from an add-on into the key business of large telecommunication companies. At the beginning of the 21st century, personal wireless communication has become part of the daily life of most people in developed areas. Together with the daily life usage, the service provided by the telecommunication companies is evolving from voice-based telephony to more demanding multimedia service, including email, web browsing, database access, video on demand, video conferencing, remote sensing, and medical applications. Multimedia services require much higher data rates than voice-centered service and they make the design of future wireless communication networks ever more challenging.

Cross-layer design was proposed to address those challenges. The concept of cross-layer design introduces interlayer coupling across the protocol stack and allows the exchange of necessary information between different layers. Although this concept can be employed in all communication networks, it is especially important in wireless networks because of the unique challenges of the wireless environment, like the time-varying and the fading nature of the wireless channels. This wireless nature and user mobility lead to random variation in network performance and connectivity.

On the other hand, the introduction of independent layers has proven to be a robust and efficient design approach, and has served extremely well in the development and implementation of both past and current communication systems. The interlayer dependencies which are introduced by the proposed cross-layer design should therefore be kept to a minimum, to preserve the layered structure as much as possible. It is important that cross-layer design does not run at cross-purposes with sound and long-term architectural principles of existing communication systems [1].

In this paper, we exploit the interlayer coupling of a cross-layer design concept for streaming video delivery in a multiuser wireless environment. We focus on a cross-layer optimization between application layer, data link layer, and physical layer. Our aim is to optimize the end-to-end quality of the wireless streaming video application as well as efficiently utilizing the wireless resources. To achieve this aim, an architecture for the joint layer optimization is proposed, which provides a potential solution for the implementation of the cross-layer optimization concept. This architecture does not require a redesign of the existing protocols, but may require extra modules to implement the function of the joint optimization.

The proposed architecture is general and consists of the process of parameter abstraction, a cross-layer optimizer, and the process of decision distribution. It is designed with the goal of increasing compatibility and stability, and the goal of
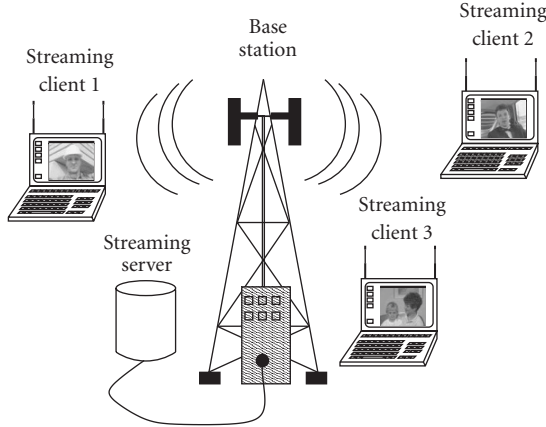
FIGURE 1: Streaming video server and mobile clients in a wireless multiuser environment.

FIGURE 2: Proposed system architecture: parameter abstraction, cross-layer optimization, and decision distribution.

reducing the signaling overhead. Every part in this architecture is formalized and its performance potential is demonstrated by sample numerical results.

An important issue in cross-layer design is the amount of the required information exchange between the layers and the time scale at which the optimization is performed. In general, the lower the amount of information exchange and the longer the time-scales are, the more robust and implementable the design becomes. In this way, our proposed cross-layer optimization interacts with the lower layers (data link layer and physical layer) on a long-term basis. This long-term approach can be extended also to the higher layers as shown in [2, 3]. This long-term approach has recently been successfully applied in [4].

There is plenty of research activity currently going on in the field of cross-layer design focusing on the interaction between physical, data link, and higher layers, sometimes also including the application layer. A review of some of these current research activities can be found in [5, 6].

In this paper, we focus on the joint optimization of three layers in the protocol stack, namely the application layer (layer 7), the data link layer (layer 2), and the physical layer (layer 1). We include the application layer in the joint optimization because the end-to-end quality observed by the users directly depends on the application and the application layer has firsthand information about the impact of each successfully decoded piece of media data on the perceived quality. We also include the physical layer and the data link layer in our consideration because the unique challenge of mobile wireless communication results from the nature of the wireless channel, which these two layers have to cope with. The main contribution of this work includes the following:

(1) *possible architecture for cross-layer optimization* which provides a potential solution of joint optimization of the physical, data link, and application layer;

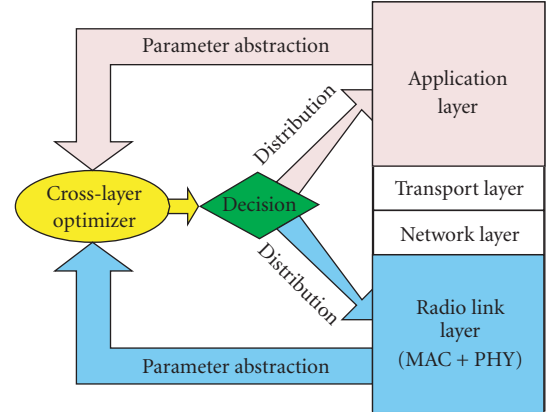(2) *mathematical description* of the proposed architecture and optimization;

(3) *simulation results* which show the possible gains that could be achieved with the proposed optimization architecture and scheme.

The structure of this paper is as follows. In Section 2, the system architecture under consideration is introduced. Then, Sections 3, 4, and 5 present the formalism of the three components in the proposed optimization architecture, respectively. We provide numerical results in Section 6 which demonstrate the potential of the proposed joint optimization. Finally, we conclude our work and discuss some further research in Section 7.

2. SYSTEM ARCHITECTURE

We consider a video streaming server located at the base station[1] and multiple streaming clients located in mobile devices. As shown in Figure 1, streaming clients or users are assumed to be sharing the same air interface and network resources but requesting different video contents. Note that only the protocol stack necessary for the wireless connection has to be considered since in our scenario the video streaming server is located directly at the base station. Therefore, the transport layer and the network layer in the protocol stack can be excluded from our optimization problem. We focus on the interaction between the application layer and the radio link layer, which incorporates both the physical (PHY) layer and the data link layer.

At the base station, an architecture as shown in Figure 2 is proposed to provide end-to-end quality-of-service optimization. This figure illustrates information flows and the tasks required for the joint optimization. The tasks can be split into three main subtasks.

(1) *Parameter abstraction:* necessary state information is collected from the application layer and the radio link layer

---

[1] Alternatively, we assume a proxy server is installed at the base station, in case the streaming server is remote.

through the process of parameter abstraction. The process of parameter abstraction results in the transformation of layer-specific parameters into parameters that are comprehensible for the cross-layer optimizer, so-called cross-layer parameters.

(2) *Cross-layer optimization:* the optimization is carried out by the cross-layer optimizer with respect to a particular objective function. From a given set of possible cross-layer parameter tuples, the tuple optimizing the objective function is selected.

(3) *Decision distribution:* after the decision on a particular cross-layer parameter tuple is made, the optimizer distributes the decision information back to the corresponding layers.

Note that an excellent discussion of other architectures, so-called top-down and bottom-up approaches, can be found in [3]. In the following, the necessity and the details of the *parameter abstraction* will be provided in Section 3, while the *cross-layer optimization* and *decision distribution* are covered in Sections 4 and 5, respectively.

3. PARAMETER ABSTRACTION

In order to carry out the joint optimization, state information or a set of key parameters have to be abstracted from the selected layers and provided to the cross-layer optimizer. This is necessary because the direct exchange of layer-specific parameters may be difficult because of the following reasons.

(1) *Compatibility:* layer-specific parameters may easily be incomprehensible or of no use for other layers. For instance, a fading correlation matrix which is meaningful at the PHY layer may well have no meaning at any of the higher layers. Its influence on system performance therefore has to be abstracted into a form which is meaningful for the other layers involved in the cross-layer optimization.

(2) *Signaling overhead:* cross-layer design requires additional signaling between the layers, which produces access delays. A reduction of the number of parameters which needs to be exchanged is therefore most welcomed. Abstraction of layer parameters can help in achieving this reduction by mapping several layer parameters into just a few abstracted parameters.

(3) *Stability:* cross-layer design introduces coupling between otherwise independent layers. Because of the latency time required in interlayer signaling, the system may become unstable. Abstraction of layer parameters can facilitate stability analysis as a consequence of the reduction of signaling overhead and the increase of compatibility. The number of the parameters is reduced and their influence on the individual layer performance may be better understood than those of the original layer parameters.

In wireless networks, the physical layer and the data link layer are dedicatedly designed for the dynamic variation of the wireless channel during the provision of a particular service. This is in contrast to wireline networks which experience much less dynamic variation. The physical layer deals with the issues including transmit power (through transmit power control), channel estimation, synchronization, signal shaping, modulation and signal detection (through signal processing), while the data link layer is responsible for radio resource allocation (multiuser scheduling or queuing) and error control (by channel coding, usually a combination of forward error-correction coding (FEC) and automatic retransmission (ARQ)). Since both of these two layers are closely related to the unique characteristics of the wireless nature, it is useful to consider them together. In the following, we refer to their combination as the radio link layer.

The application layer is the layer where the media data is compressed, packetized, and scheduled for transmission. The key parameters to be abstracted for the cross-layer optimization are related to the characteristics of the compressed source data. This implies that these abstracted key parameters may depend on the type of application or service because the characteristics of the compressed source data may depend on the application or service. In this paper, we consider a video streaming service application.

3.1. Data link layer and physical layer parameters

To formalize the data link layer and physical layer parameter abstractions, we follow the approach proposed in [7, 8] and define the set

$$\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots\} \quad (1)$$

of tuples $\mathbf{r}_i = (r_i^1, r_i^2, \dots)$ of radio-link-layer-specific parameters r_i^j (e.g., modulation alphabets, code rate, airtime, transmit power, decorrelation time). Since these radio-link-layer-specific parameters may be variable, the set $\mathcal{R}$ contains all possible combinations of their values and each tuple $\mathbf{r}_i$ represents one possible combination. In this way, $\mathcal{R}$ can be an infinite, countably infinite, or finite set, depending on the discrete or continuous nature of the parameter tuples. In order to formalize parameter abstraction, we define the set

$$\widetilde{\mathcal{R}} = \{\widetilde{\mathbf{r}}_1, \widetilde{\mathbf{r}}_2, \dots\} \quad (2)$$

of tuples $\widetilde{\mathbf{r}}_i = (\widetilde{r}_i^1, \widetilde{r}_i^2, \dots)$ of abstracted parameters. The relationship between the set $\mathcal{R}$ of all possible radio link layer parameter tuples and the set $\widetilde{\mathcal{R}}$ of all possible *abstracted* radio link layer parameter tuples is established by the relation

$$\mathbb{G} \subseteq \mathcal{R} \times \widetilde{\mathcal{R}} \quad (3)$$

with domain $\mathcal{R}$ and codomain $\widetilde{\mathcal{R}}$. Here, the symbol $\times$ refers to the Cartesian product. The relation $\mathbb{G}$ is a subset of $\mathcal{R} \times \widetilde{\mathcal{R}}$ that defines the mapping between $\mathcal{R}$ and $\widetilde{\mathcal{R}}$. That is, only and all valid pairs $(\mathbf{r}_i, \widetilde{\mathbf{r}}_j)$ are elements of $\mathbb{G}$. We call this mapping process the *radio link layer parameter abstraction*.

Let us look at an example. In a single-user scenario, we could, for example, abstract four key parameters: transmission data rate d, transmission packet error ratio e, data packet size s, and the channel decorrelation time t. This leads to the abstracted parameter tuple $\widetilde{\mathbf{r}}_i = (d_i, e_i, s_i, t_i)$. In a K user scenario, one can extend the parameter abstraction for each user. The parameter tuple $\widetilde{\mathbf{r}}_i$ then contains $4K$ parameters,
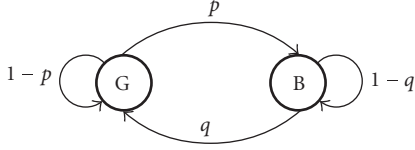
FIGURE 3: A two-state Markov channel model.

$\widetilde{\mathbf{r}}_i = (d_i^{(1)}, e_i^{(1)}, s_i^{(1)}, t_i^{(1)}, \ldots, d_i^{(K)}, e_i^{(K)}, s_i^{(K)}, t_i^{(K)})$, in which a group of four parameters belongs to one user. The transmission data rate d is influenced by the modulation scheme, the code rate of the used channel code, and the multi-user scheduling. The transmission packet error ratio e is influenced by the transmit power, channel estimation, signal detection, modulation scheme, channel coding, the current user position, and so forth, The channel decorrelation time t of a user is related to the user's velocity and its surrounding environment, while the data packet size s is normally defined by the wireless system standard. These interrelationships define the relation $\mathbb{G}$ from (3). A detailed discussion of the relation $\mathbb{G}$ can be found in [2].

Alternatively, it is possible to transform the transmission packet error ratio e and the channel decorrelation time t into the two parameters of a two-state Markov model as shown in Figure 3, which are the transition probabilities (p and q) from one state to another. In Figure 3, the states G and B represent the good and bad states, respectively. The transformation is given by [9] as

$$p = \frac{es}{td}, \qquad q = \frac{(1-e)s}{td}, \tag{4}$$

where p is the transition probability from the good to the bad state and q is the transition probability from the bad to the good state. In this way, the abstracted parameter tuples take on the form $\widetilde{\mathbf{r}}_i = (d_i^{(1)}, s_i^{(1)}, p_i^{(1)}, q_i^{(1)}, \ldots, d_i^{(K)}, s_i^{(K)}, p_i^{(K)}, q_i^{(K)})$. One advantage of this transformation is that the resulting parameter tuple is more comprehensible for high layers in the protocol stack.

3.2. Application layer parameters

Similar to the parameter abstraction in Section 3.1, for a formal description, let us define the set

$$\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \} \tag{5}$$

of tuples $\mathbf{a}_i = (a_i^1, a_i^2, \ldots,)$ of application-layer-specific parameters a_i^j. Since these application-layer-specific parameters may be variable, the set $\mathcal{A}$ contains all possible combinations of their values and each tuple $\mathbf{a}_i$ represents one possible combination. We further define the set $\widetilde{\mathcal{A}} = \{\widetilde{\mathbf{a}}_1, \widetilde{\mathbf{a}}_2, \ldots\}$ of tuples $\widetilde{\mathbf{a}}_i = (\widetilde{a}_i^1, \widetilde{a}_i^2, \ldots)$ of abstracted parameters $\widetilde{a}_i^j$. The relationship between $\mathcal{A}$ and $\widetilde{\mathcal{A}}$ is established by the relation

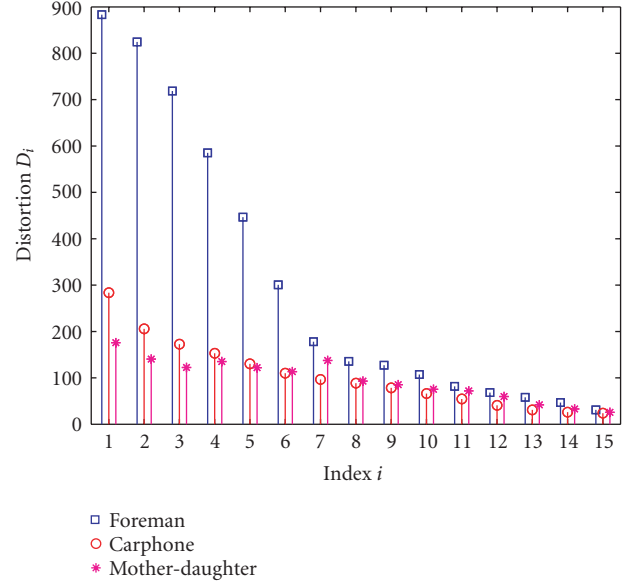$$\mathbb{F} \subseteq \mathcal{A} \times \widetilde{\mathcal{A}} \tag{6}$$

FIGURE 4: Measured loss distortion profile for a GOP in three video sequences.

with domain $\mathcal{A}$ and codomain $\widetilde{\mathcal{A}}$. The relation $\mathbb{F}$ is a subset of $\mathcal{A} \times \widetilde{\mathcal{A}}$ that defines the mapping between $\mathcal{A}$ and $\widetilde{\mathcal{A}}$. That is, only and all valid pairs $(\mathbf{a}_i, \widetilde{\mathbf{a}}_j)$ are elements of $\mathbb{F}$. We call this mapping process the *application layer parameter abstraction*.

In this paper, we assume a streaming video service. The abstracted parameters of this service include the source data rate, the number of frames (or pictures) per second, size (in terms of bytes), and maximum delay of each frame (or picture). Other important information for the optimizer is the distortion-rate function (encoding distortion) and the so-called *loss distortion profile*, which shows the distortion D_i that is introduced in case the ith frame of the GOP is lost. Figure 4 shows an example of the loss distortion profile of lost frames for three different video sequences. This profile is generated from a group of picture (GOP) with 15 frames, starting with an independently decodable intraframe and followed by 14 interframes. The interframes can only be successfully decoded if all previous frames of the same GOP are decoded error-free. The index in Figure 4 indicates the loss of a particular frame, while the distortion D_i is quantified by the mean-squared reconstruction error (MSE), which is measured between the displayed and the transmission error-free decoded video sequence. It is assumed that as part of the error concealment strategy, all the following frames of the GOP are not decodable and the most recent correctly decoded frame is displayed instead of the nondecoded frames (copy the previous frame error concealment).

3.3. Cross-layer parameters

The abstracted parameter sets ($\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{A}}$) from both the application layer and the radio link layer form the input to the cross-layer optimizer. Since any combination of the abstracted parameter tuples from the two input sets is valid, it

is convenient to define the cross-layer parameter set

$$\widetilde{\mathcal{X}} = \widetilde{\mathcal{R}} \times \tilde{\mathcal{A}} \qquad (7)$$

which combines the two input sets into one input set for the optimizer. The set $\widetilde{\mathcal{X}} = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots\}$ consists of tuples $\tilde{\mathbf{x}}_n = (\tilde{\mathbf{r}}_i, \tilde{\mathbf{a}}_j)$. Note that the cardinality of the set $\widetilde{\mathcal{X}}$ grows exponentially with the number of cross-layer parameters.[2] This means that the complexity of the cross-layer optimization grows exponentially with the number of cross-layer parameters.

4. THE CROSS-LAYER OPTIMIZER

With the formalism introduced in Section 3, the operation of the cross-layer optimizer Ω can now be described by

$$\Omega : \widetilde{\mathcal{X}} \longrightarrow \widetilde{\mathcal{Y}} \subset \widetilde{\mathcal{X}}. \qquad (8)$$

The optimizer gets as input the set $\widetilde{\mathcal{X}}$ of all possible abstracted cross-layer parameter tuples and returns a true nonempty subset $\widetilde{\mathcal{Y}}$ as its output. In the following, we assume that $|\widetilde{\mathcal{Y}}| = 1$, that is, the output of the optimizer is a single tuple and

$$\widetilde{\mathcal{Y}} = \{\tilde{\mathbf{x}}_{\mathrm{opt}}\} \in \widetilde{\mathcal{X}}. \qquad (9)$$

The decision or output $\tilde{\mathbf{x}}_{\mathrm{opt}}$ of the cross-layer optimizer is made with respect to a particular objective function

$$\Gamma : \widetilde{\mathcal{X}} \longrightarrow \mathbb{R}, \qquad (10)$$

where $\mathbb{R}$ is the set of real numbers. Therefore, the output of the optimizer can be expressed as

$$\tilde{\mathbf{x}}_{\mathrm{opt}} = \arg\min_{\tilde{\mathbf{x}} \in \widetilde{\mathcal{X}}} \Gamma(\tilde{\mathbf{x}}). \qquad (11)$$

Notice that because $\widetilde{\mathcal{X}}$ is a finite set, the optimization (11) is performed by exhaustive search guaranteeing the global optimal solution. The choice of a particular objective function Γ depends on the goal of the system design, and the output (or decision) of the optimizer might be different for different objective functions. In the example application of streaming video, one possible objective function in a single-user scenario is the MSE between the displayed and the original video sequence, that is, the sum of loss distortion MSE_L and source distortion MSE_S:

$$\mathrm{MSE} = \mathrm{MSE}_S + \mathrm{MSE}_L, \qquad (12)$$

where MSE_L can be computed from the distortion profile by

$$\mathrm{MSE}_L = \sum_{i=1}^{15} D_i P_i, \qquad (13)$$

where P_i is the probability that the ith frame is the first frame lost during transmission of this GOP and D_i is the mean-square error that is introduced by this loss. Note that the D_i is taken from the measured distortion profile and is usually different for each GOP. Figure 4 shows an example distortion profile. The P_i can be computed from the 2-state Markov model as shown in Figure 3. For details, we refer to [2, 10, 11].

For a multiuser situation, different extensions of the MSE are possible. For example, the objective function can be the sum of MSE of all the users. That is,

$$\Gamma(\tilde{\mathbf{x}}) = \sum_{k=1}^{K} \mathrm{MSE}_k(\tilde{\mathbf{x}}), \qquad (14)$$

where $\mathrm{MSE}_k(\tilde{\mathbf{x}})$ is the MSE of user k for the cross-layer parameter tuple $\tilde{\mathbf{x}} \in \widetilde{\mathcal{X}}$. This objective function will optimize the average performance in terms of MSE among all users. Another common definition of the objective function is

$$\Gamma(\tilde{\mathbf{x}}) = \max_{k \in \{1, 2, \ldots, K\}} \mathrm{MSE}_k(\tilde{\mathbf{x}}) \qquad (15)$$

which ensures that the MSE is minimized with the constraint that all users obtain the same MSE.[3] Yet another definition

$$\Gamma(\tilde{\mathbf{x}}) = \prod_{k=1}^{K} \mathrm{MSE}_k(\tilde{\mathbf{x}}) \qquad (16)$$

leads to a maximization of the average PSNR among all users.

5. DECISION DISTRIBUTION

Once the output $\tilde{\mathbf{x}}_{\mathrm{opt}} = (\tilde{\mathbf{r}}_{\mathrm{opt}}, \tilde{\mathbf{a}}_{\mathrm{opt}})$ of the cross-layer optimizer is obtained, the decisions $\tilde{\mathbf{r}}_{\mathrm{opt}} \in \widetilde{\mathcal{R}}$ and $\tilde{\mathbf{a}}_{\mathrm{opt}} \in \tilde{\mathcal{A}}$ have to be communicated back to the radio link layer and the application layer, respectively. During this, the process of parameter abstraction has to be reversed and the abstracted parameters $\tilde{\mathbf{r}}_{\mathrm{opt}}$ and $\tilde{\mathbf{a}}_{\mathrm{opt}}$ have to be transformed back to the layer-specific parameters $\mathbf{r}_{\mathrm{opt}} \in \mathcal{R}$ and $\mathbf{a}_{\mathrm{opt}} \in \mathcal{A}$. This reverse transformation is given by

$$\begin{aligned} \mathbf{r}_{\mathrm{opt}} &\in \{\mathbf{r} \mid (\mathbf{r}, \tilde{\mathbf{r}}_{\mathrm{opt}}) \in \mathbb{G}\}, \\ \mathbf{a}_{\mathrm{opt}} &\in \{\mathbf{a} \mid (\mathbf{a}, \tilde{\mathbf{a}}_{\mathrm{opt}}) \in \mathbb{F}\}. \end{aligned} \qquad (17)$$

In case that $\{\mathbf{r} \mid (\mathbf{r}, \tilde{\mathbf{r}}_{\mathrm{opt}}) \in \mathbb{G}\}$ or $\{\mathbf{a} \mid (\mathbf{a}, \tilde{\mathbf{a}}_{\mathrm{opt}}) \in \mathbb{F}\}$ has more than one element, the choice of particular elements $\mathbf{r}_{\mathrm{opt}}$ and $\mathbf{a}_{\mathrm{opt}}$, respectively, can be made at the corresponding layers individually.

6. SAMPLE NUMERICAL RESULTS

In this section, we provide sample simulation results to evaluate the performance of the proposed joint optimization. We

---

[2] For instance, assume that all, say n, cross-layer parameters are quantized to a fixed number, say q, of values. Then the cardinality of the set $\widetilde{\mathcal{X}}$ becomes q^n, which shows exponential growth in the number of cross-layer parameters.

[3] In practice, some or all of the cross-layer parameters may only take on values from a finite set. The resulting granularity in general leads to not all users having the same quality of service as would be the case if all parameters were continuously adjustable.

TABLE 1: Multiuser scheduling: TDMA airtime assignment.

Case →	1	2	3	4	5	6	7
User 1	3/9	4/9	4/9	3/9	2/9	3/9	2/9
User 2	3/9	3/9	2/9	4/9	4/9	2/9	3/9
User 3	3/9	2/9	3/9	2/9	3/9	4/9	4/9

TABLE 2: Resulting transmission data rates in kbps for each user.

Case →	1	2	3	4	5	6	7	Mod.
User 1	150	200	200	150	100	150	100	
User 2	150	150	100	200	200	100	150	BPSK
User 3	150	100	150	100	150	200	200	
Case →	8	9	10	11	12	13	14	Mod.
User 1	300	400	400	300	200	300	200	
User 2	300	300	200	400	400	200	300	QPSK
User 3	300	200	300	200	300	400	400	

assume $K = 3$ users or clients (users 1, 2, and 3), each of which requests a different video sequence. We assume that users 1, 2, and 3 request the carphone (CP), foreman (FM), and mother & daughter (MD) video test sequence, respectively.[4]

6.1. Objective function

We choose the peak-signal-to-noise ratio (PSNR) as our performance measure. The PSNR is defined as

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{255^2}{\text{MSE}}\right). \tag{18}$$

The larger the PSNR is, the smaller the MSE is, which is computed between the original video sequence and the reconstructed sequence at the client or user. Therefore, the larger the PSNR is, the better the performance is. As an example, we use the objective function given in (15), which maximizes the worst-case user's performance. Therefore, the cross-layer optimizer chooses the parameter tuple that minimizes the maximum of MSE (or equivalently maximizes the minimum of the PSNR) among the users. This leads to all users having the same PSNR. However, the PSNR may nevertheless come out different for each user because of granularity of the cross-layer parameters (see footnote 3).

6.2. Physical layer and data link layer parameters

In the simulation, it is assumed that the data packet size s at the radio link layer equals 432 bits, which is the same as the specified packet size of the *IEEE*802.11*a* or HiperLAN2 standard [12]. The channel decorrelation time t is assumed to be 50 milliseconds for all the three users, which corresponds to a pedestrian speed (about 2 Km/h at 5 GHz carrier frequency).

Since the transmission data rate d is influenced by modulation scheme, the channel coding, and the multiuser scheduling, two different modulations (BPSK and QPSK) are assumed. It is further assumed that there are 7 cases

---

[4] We have chosen these particular video test sequences as they emphasize different situations in a real-world video sequence. FM contains a scene change with rather quick camera movement, MD has no camera movement or scene change, while CP has a quickly moving background accompanied by medium foreground movement. These situations typically occur in real-life video sequences and lead to rather different properties of the encoded data streams, especially bit sizes of frames and sensitivity to frame losses.

of time arrangement in a time-division multiplexing-based multiuser scheduling as shown in Table 1. A user's transmission data rate is assumed to be equal to 100 kbps provided that BPSK is used and 2/9 of the total transmission time is assigned to it. Therefore, if QPSK is used and 4/9 of the total transmission time is assigned, the user can have a transmission data rate as high as 400 kbps. Table 2 shows the resulting transmission rate for each user as a function of the time arrangement and modulation scheme (BPSK or QPSK).

The transmission error rate on the other hand depends on the transmission data rate, the average SNR, and the error-correcting capability of the channel code. Usually, the performance of a channel code is evaluated in terms of the residual error rate (after channel decoding) for a given receive SNR. In our simulation, we assume a convolutional code of code rate 1/2 and a data packet size of 432 bits. The residual packet error ratio is shown in Figure 5(a) as a function of SNR [12]. However, in the wireless link, the receive SNR is not constant, but is fluctuating around the mean value (long-term SNR), which is due to fast fading caused by user mobility. In this way, the receive SNR can be modeled as a random variable with a certain probability distribution, which is determined by the propagation property of the physical channel (e.g., Rayleigh distribution, Rice distribution). The residual packet error rate in a fading wireless link is computed by averaging this packet error ratio (e.g., taken from Figure 5(a)) with the fading statistics. Assuming Rayleigh fading, the resulting average packet error rate is given in Figure 5(b) as a function of the *average* signal-to-noise ratio $\overline{\text{SNR}}$. This resulting average packet error ratio is used as the parameter e in (4) in our simulation.

User's position-dependent path loss and shadowing commonly observed in wireless links are taken into account by choosing the long-term average SNR randomly and independently for each user uniformly within the range from 1 to 100 (0 dB to 20 dB).

In summary, the abstracted parameters, namely date rate d_i, packet size s_i, and Markov model parameters (p_i, q_i) for each user and each of the 7 or 14 cases of modulation and TDMA scheduling scheme (according to Table 1 or 2, resp.), have to be communicated to the cross-layer optimizer.
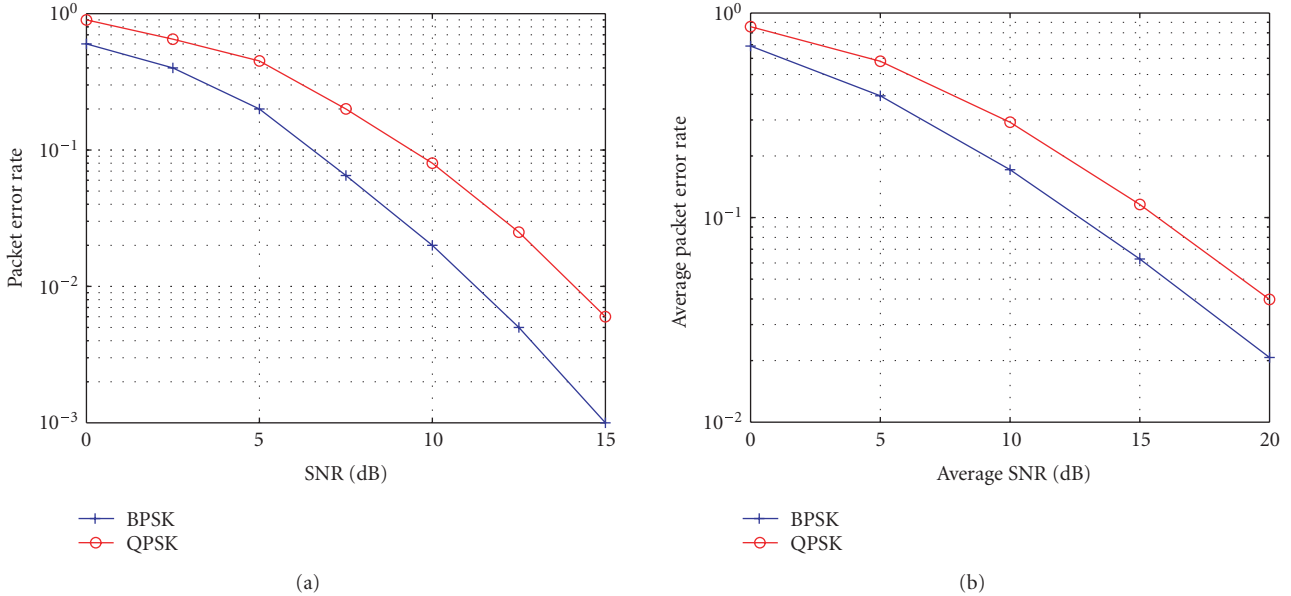
FIGURE 5: Example decoding error performances of a convolutional code with different modulations in an AWGN and a Rayleigh fading channels: (a) packet error ratio after channel decoding as a function of the signal-to-noise ratio (SNR) in an AWGN channel [12]; (b) packet error ratio after channel decoding as a function of the *average* signal-to-noise ratio $\overline{\text{SNR}}$ in a *Rayleigh fading* channel.

6.3. Application layer parameters

At the application layer, it is assumed that the video is encoded using the $H.264/AVC$ [13] video compression standard with 30 frames per second and 15 frames per GOP (i.e., 0.5-second GOP duration). Two different values of the source rate (100 kbps and 200 kbps) are considered. To this end, the video has been pre-encoded at these two different target rates and both versions are stored on the streaming server. We can switch from one source stream to the other at the beginning of any GOP. In each GOP, the first frame is an I-frame and the following 14 frames are P-frames. We use the measured distortion profile of a particular lost frame and the encoding distortion for the 3 requested videos. Figure 4 shows an example of a distortion profile in terms of MSE for a GOP at a source rate of 100 kbps. Also, note that since successful decoding of P-frames depends on error-free reception of all previous frames of the same GOP, losing the first frame of a GOP leads to the largest distortion, while losing the last frame of a GOP leads to the least distortion. Furthermore, it is assumed that each video frame (or picture) is packetized with maximum size of 432 bits and each packet only contains data from one frame. The size of each frame is determined during the H.264/AVC encoding. These values are stored along with the bit stream and the distortion profile as well as the value of the source distortion. Table 3 gives the measured size (in terms of packets) for a GOP in the three video sequences at a source rate of 100 kbps, where I and Pn ($n = 1, 2, \ldots, 14$) denote the I-frame and the nth P-frame, respectively. We can see that the size of an I-frame is much larger than that of a P-frame and the size of a P-frame varies from frame to frame. This is related to the contents of a video.

In summary, the abstracted parameters, namely the loss distortion profile as shown in Figure 4 and the frame sizes as shown Table 3 for each user, have to be communicated to the cross-layer optimizer.

6.4. Operating modes

An operation mode without ARQ (referred to as forward mode) and an operation mode with ARQ (referred to as ARQ mode) are investigated. We consider every GOP as a unit and assume that each GOP has to be transmitted within the duration of 0.5 second.

(i) *Forward mode:* we assume no acknowledgment from the clients is available and the video frames of every GOP for a particular client are repeatedly transmitted when the transmission data rate is larger than the source data rate. For instance, every GOP is transmitted twice if the transmission data rate is twice as large as the source data rate. If the transmission data rate is 1.5 times the source data rate, a GOP is transmitted once followed by retransmitting the I-frame, the first P-frame, the second P-frame, and so forth, until the period of 0.5 second for the GOP is expired.

(ii) *ARQ mode:* here we assume that instantaneous acknowledgment of a transmitted packet is available from the clients and the data packets of every GOP for a particular client are retransmitted in the way that the data packets in a GOP are received successfully in time order. That is, before transmitting a new packet, it is guaranteed that its previous packets in the GOP are received correctly.

In the following, both modes of operation will be investigated.

6.5. Simulation results and discussion

Figures 6 and 7 provide simulation results of the following three scenarios.

TABLE 3: Measured sizes (in number of packets) of the encoded frames of a GOP for three different video sequences at 100 kbps.

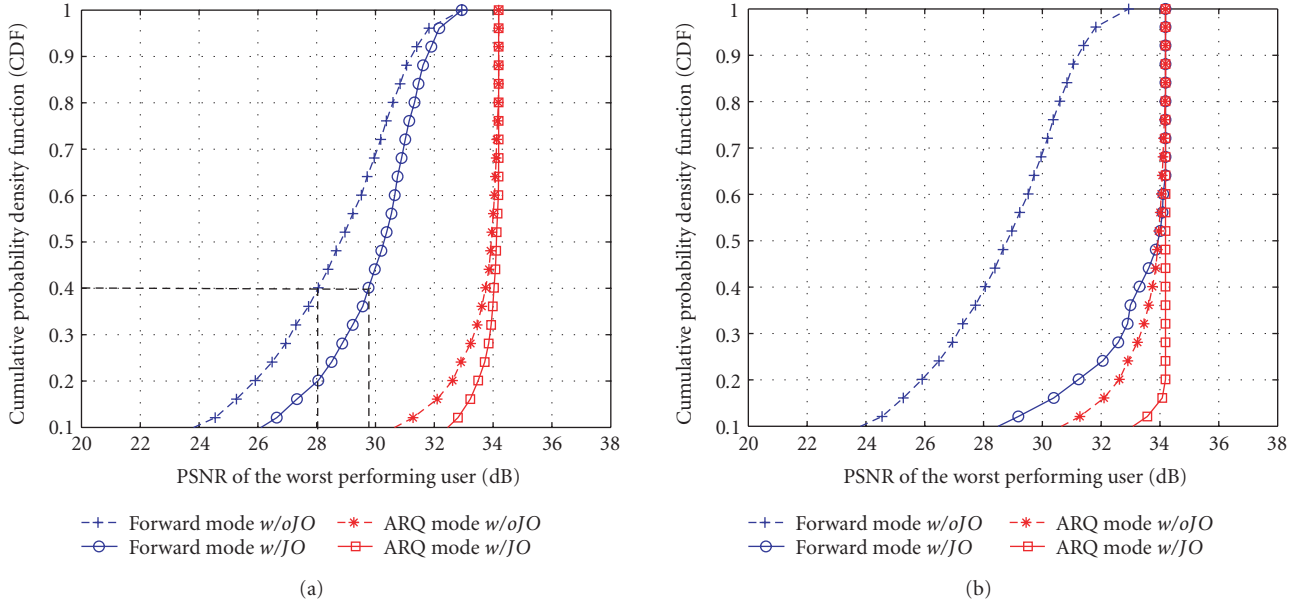| Frame → | I | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 | P13 | P14 |
Sequence ↓															
Carphone	43	7	7	7	6	5	8	6	7	7	6	6	4	5	5
Foreman	47	5	6	7	5	6	7	6	5	5	6	5	5	3	4
Mother & daughter	50	1	2	3	3	3	4	4	4	5	6	8	10	12	14

(a)

(b)

FIGURE 6: Cumulative probability density function (CDF) of the PSNR of the worst performing user: (a) results for scenario 1, BPSK modulation, and source rate of 100 kbps; (b) results for scenario 2, BPSK/QPSK modulation, and source rate of 100 kbps.

(1) *Scenario 1:* we restrict ourselves that only BPSK modulation is used at the radio link layer and only the source rate with 100 kbps is available at the application layer. Therefore, only one constant abstracted parameter tuple (with 100 kbps for all 3 users) is provided by the application layer (i.e., $|\widetilde{\mathcal{A}}| = 1$) in this scenario, while the radio link layer provides 7 abstracted parameter tuples ($|\widetilde{\mathcal{R}}| = 7$), which result from the 7 cases of time arrangement shown in Table 1. The cross-layer optimizer selects one out of the 7 combinations of the input parameter tuples ($|\widetilde{\mathcal{X}}| = |\widetilde{\mathcal{R}}| \cdot |\widetilde{\mathcal{A}}| = 7$) such that our objective function given in (15) is optimized.

(2) *Scenario 2:* the same abstracted parameter tuple as in scenario 1 is assumed at the application layer but the radio link layer provides 14 abstracted parameter tuples, which result from the 7 cases of time arrangement with BPSK and another 7 cases of time arrangement with QPSK.

(3) *Scenario 3:* it is assumed that the two different source rates of 100 kbps and 200 kbps for each of the 3 users are provided by the application layer. This results in $|\widetilde{\mathcal{A}}| = 2^3 = 8$ abstracted parameter tuples from the application layer. The same 14 abstracted parameter tuples as in scenario 2 are provided by the radio link layer.

The distortion MSE given in (12) is a random variable controlled by the two factors, namely fast fading and

user's position-dependent path loss and shadowing. In general, fast fading takes place on a much smaller time scale than the path loss and shadowing. In this paper, we evaluate the MSE averaged over fast fading by taking the expected value of the MSE with respect to the fast fading for a particular position of the users or equivalently for a particular long-term SNR. Based on this value, the cross-layer optimizer makes its decision. We also look at its statistical properties for an ensemble of user positions. Therefore, the cumulative probability density function (CDF) of this average MSE is chosen to show the performance of both modes (forward mode and ARQ mode). The performance of the worst performing user in the system with the proposed joint optimization (*w/JO*) is compared with that in a system without joint optimization (*w/oJO*). A system without joint optimization is assumed to assign the same amount of transmission time to all the users (i.e., Case 1 in Table 1) and use BPSK modulation, while the source data rate is fixed to 100 kbps. It can be seen from Figure 6(a) that the PSNR of the worst performing user improves significantly in the system *w/JO*. For instance, there is about $1 - 40\% = 60\%$ chance that the PSNR of the worst performing user is larger than 30 dB in the system *w/JO* in forward mode, which improves to 2 dB when compared to the system *w/oJO*.
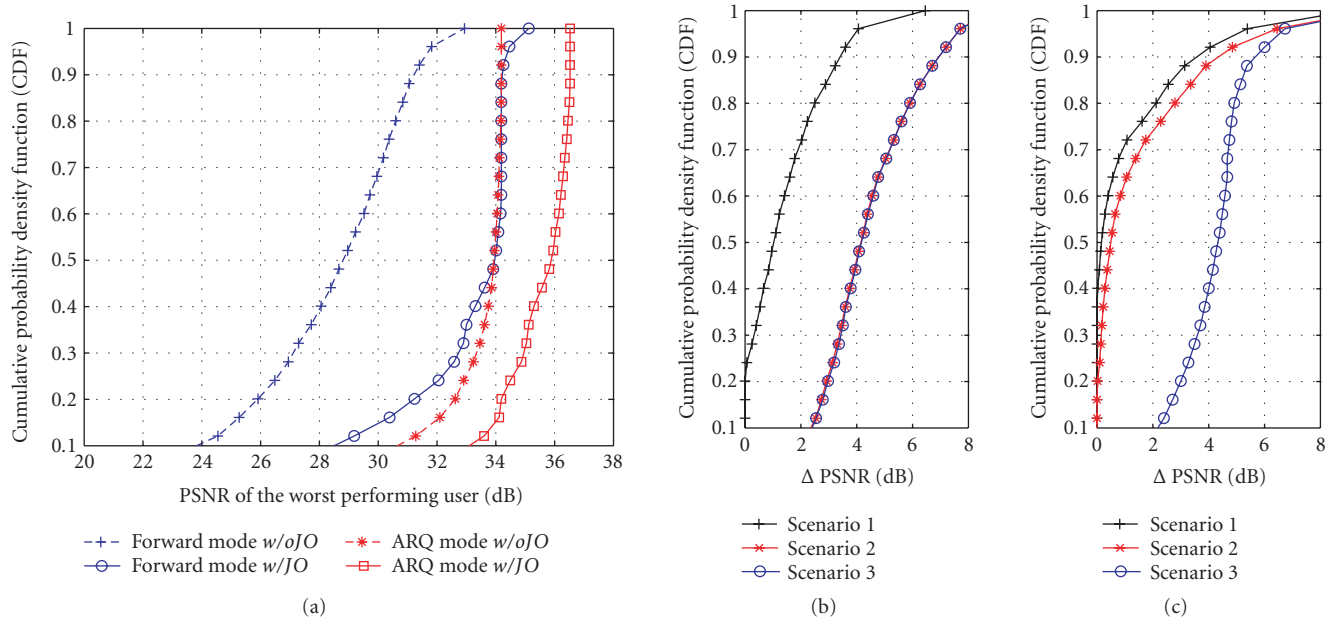
FIGURE 7: (a) Cumulative probability density function (CDF) of the PSNR of the worst performing user for scenario 3, BPSK/QPSK modulation and source rate of 100 kbps/200 kbps; (b) performance improvement for the three scenarios in *forward mode*; (c) performance improvement in *ARQ mode.*

A similar trend of improvement can be observed in Figure 6(b) and Figure 7(a) for scenarios 2 and 3, respectively. The performance improves when more abstracted parameter tuples are provided because more degrees of freedom can be obtained. This can be observed in Figure 7(b) and Figure 7(c) more clearly, where the performance improvement of the three investigated scenarios is shown. Here, ΔPSNR is defined as the difference between the PSNR of the worst performing user in the system *w/JO* and that in the system *w/oJO*. A close observation of Figure 7(b) reveals that the amount of performance improvement of scenario 2 is much larger than that of scenario 1 in forward mode, while the amount of performance improvement of scenario 3 is only slightly larger than that of scenario 2. This indicates that the choice of higher transmission data rate (by using QPSK) provided by the radio link layer is favorable in forward mode, and the optimizer chooses it frequently. In contrast, the choice of higher source rate (200 kbps) provided by the application layer is not so favorable in this mode and the optimizer seldom chooses it. On the other hand, this choice of a higher source rate is favorable in ARQ mode, which can be seen from the graph in Figure 7(b), where the amount of performance improvement of scenario 3 is fairly larger than that of scenario 2. Therefore, choosing a suitable set of abstracted parameters tuples is important in order to obtain large performance improvements while optimizing at low complexity.

7. CONCLUSION AND OUTLOOK

We have exploited the interlayer coupling of a cross-layer design concept and proposed an architecture for the joint optimization with three principle concepts, namely parameter abstraction, cross-layer optimization, and decision distribution. Although we have focused on the application layer and radio link layer in a wireless system with a video streaming service, this architecture can be easily generalized for different layers and different services. Our study reveals that this proposed architecture can provide a potential way to improve the performance and therefore help dealing with the future challenges in wireless multimedia communication. Even when considering a small number of degrees of freedom of the application layer and the radio link layer, we obtain significant improvements in user-perceived quality of our streaming video application by joint optimization. Note that we only consider the wireless hop in this study. Further sophisticated research might be required in order to exploit this cross-layer design concept more completely. This work has been partially presented at ICIP'04 [14].
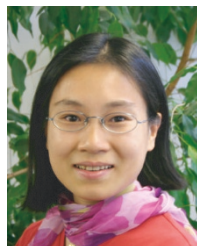
REFERENCES

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

[2] L. Choi, M. T. Ivrlač, E. Steinbach, and J. A. Nossek, "Bottom-up approach to cross-layer design for video transmission over

wireless channels," in *Proceedings of the IEEE Vehicular Technology Conference (VTC '05)*, pp. 3019–3023, Stockholm, Sweden, May 2005.

[3] M. T. Ivrlač, *Wireless MIMO Systems - Models, Performance, Optimization*, Shaker, Aachen, Germany, 2005.

[4] J. Brehmer and W. Utschick, "Modular cross-layer optimization based on layer descriptions," in *Proceedings of the Wireless Personal Multimedia Communications Symposium (WPMC '05)*, Aalborg, Denmark, September 2005.

[5] M. Van Der Schaar and S. Shankar N, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 50–58, 2005.

[6] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, "Application-driven cross-layer optimization for video streaming over wireless networks," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 122–130, 2006.

[7] M. T. Ivrlač and F. Antreich, "Cross OSI layer optimization - an equivalence class approach," Tech. Rep. TUM-LNS-TR-03-09, Institute for Circuit Theory and Signal Processing, Munich University of Technology, Munich, Germany, May 2003.

[8] M. T. Ivrlač and J. A. Nossek, "Cross layer design - an equivalence class approach," in *Proceedings of the International Symposium on Signals, Systems, and Electronics (ISSSE '04)*, Linz, Austria, August 2004.

[9] M. T. Ivrlač, "Parameter selection for the Gilbert-Elliott model," Tech. Rep. TUM-LNS-TR-03-05, Institute for Circuit Theory and Signal Processing, Munich University of Technology, Munich, Germany, May 2003.

[10] L. U. Choi, M. T. Ivrlač, E. Steinbach, and J. A. Nossek, "Analysis of distortion due to packet loss in streaming video transmission over wireless communication links," in *Proceedings of the International Conference on Image Processing (ICIP '05)*, vol. 1, pp. 189–192, Genova, Italy, September 2005.

[11] Y. Peng, S. Khan, E. Steinbach, M. Sgroi, and W. Kellerer, "Adaptive resource allocation and frame scheduling for wireless multi-user video streaming," in *Proceedings of the International Conference on Image Processing (ICIP '05)*, vol. 3, pp. 708–711, Genova, Italy, September 2005.

[12] J. Khun-Jush, G. Malmgren, P. Schramm, and J. Torsner, "HIPERLAN type 2 for broadband wireless communication," *Ericsson Review*, vol. 77, no. 2, pp. 108–119, 2000.

[13] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[14] L. Choi, W. Kellerer, and E. Steinbach, "Cross layer optimization for wireless multi-user video streaming," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, vol. 3, pp. 2047–2050, Singapore, Republic of Singapore, October 2004.

Lai-U Choi received the B.Eng. degree from the University of Macau, Macau, in 1998. She was educated in the Hong Kong University of Science and Technology (HKUST), Hong Kong, for the M.Phil. and the Ph.D. study from 1998 to 2003, all in electrical and electronic engineering. During this period, she has also been a Research Assistant conducting research on MIMO signal processing for downlink wireless communications at HKUST. After she obtained her Ph.D. degree in 2003, she has joined the Department of Electrical Engineering and Information Technology at Munich University of Technology, Germany. Her current research interests include the areas of smart/MIMO antenna systems, multiuser communications, signal processing for wireless communications, multimedia communications, communication networks, resource allocation, and coding theory.

Wolfgang Kellerer is a Senior Manager at NTT DoCoMo's European Research Laboratories, Munich, Germany, heading the Ubiquitous Services Platform Research Unit. His current research interests are in the area of mobile systems focusing on mobile service platforms, peer-to-peer, sensor networks, and cross-layer design. In 2004 and 2005, he has served as the elected Vice Chairman of the Working Group 2 (Service Architecture) of the Wireless World Research Forum (WWRF). He is a Member of the editorial board of Elsevier's International Journal of Computer and Telecommunications Networking (COMNET) and serves as a Guest Editor for the IEEE Communications Magazine in 2006. He has published over 60 papers in respective journals, conferences, and workshops in the area of service platforms and mobile networking and he filed more than 20 patents. Before he joined DoCoMo Euro-Labs, he has been a Member of the research and teaching staff at the Institute of Communication Networks at Munich University of Technology. In 2001, he was a Visiting Researcher at the Information Systems Laboratory of Stanford University. He received a Dipl.-Ing. degree (M.S.) and a Dr.-Ing. (Ph.D.) degree in electrical engineering and information technology from Munich University of Technology, Germany, in December 1995 and in January 2002, respectively. He is a Member of IEEE ComSoc and the German VDE/ITG.

Eckehard Steinbach studied electrical engineering at the University of Karlsruhe (Germany), the University of Essex (Great Britain), and Ecole Supérieme d' Ingénieurs en Électronique et Électrotechnique (ESIEE) in Paris. From 1994 to 2000, he was a Member of the research staff of the Image Communication Group at the University of Erlangen-Nuremberg (Germany), where he received the Engineering Doctorate in 1999. From February 2000 to December 2001, he was a postdoctoral fellow with the Information Systems Laboratory of Stanford University. In February 2002, he joined the Department of Electrical Engineering and Information Technology of Munich University of Technology (Germany), where he is currently an Associate Professor for Media Technology. His current research interests are in the area of networked and interactive multimedia systems. He served as a Conference Cochair of "SPIE Visual Communications and Image Processing (VCIP)" in San Jose, Calif, in 2001, and "Vision, Modeling and Visualization 2003 (VMV)" in Munich, in November 2003. He has been a Guest Editor of the Special Issue on Multimedia over IP and Wireless Networks of the EURASIP Journal on Applied Signal Processing. He currently is a Guest Editor of the EURASIP Journal on Applied Signal Processing, Special Issue on Advanced Video Technologies and Applications for H.264/AVC and Beyond. From 2006 to 2007, he serves as an Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technology (CSVT).