

## Research Article

# MOS-Based Multiuser Multiapplication Cross-Layer Optimization for Mobile Multimedia Communication

Shoaib Khan,<sup>1</sup> Svetoslav Duhovnikov,<sup>1</sup> Eckehard Steinbach,<sup>1</sup> and Wolfgang Kellerer<sup>2</sup>

<sup>1</sup>Media Technology Group, Technische Universität München, 80687 München, Munich, Germany

<sup>2</sup>DoCoMo Communications Laboratories Europe GmbH, Future Networking Lab, 80687 2 Munich, Germany

Received 7 January 2007; Revised 30 April 2007; Accepted 5 June 2007

Recommended by Jianwei Huang

We propose a cross-layer optimization strategy that jointly optimizes the application layer, the data-link layer, and the physical layer of a wireless protocol stack using an application-oriented objective function. The cross-layer optimization framework provides efficient allocation of wireless network resources across multiple types of applications run by different users to maximize network resource usage and user perceived quality of service. We define a novel optimization scheme based on the mean opinion score (MOS) as the unifying metric over different application classes. Our experiments, applied to scenarios where users simultaneously run three types of applications, namely voice communication, streaming video and file download, confirm that MOS-based optimization leads to significant improvement in terms of user perceived quality when compared to conventional throughput-based optimization.

Copyright © 2007 Shoaib Khan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

In order to achieve efficient resource usage in a wireless network and to provide high quality of services to the largest possible number of users, it is necessary to obtain an optimal configuration of the wireless transmission system. Dynamic changes of transmission conditions and concurrently running applications by different users make dynamic optimization of resources a complex task. In realistic scenarios, multiple users share the wireless medium and run rather diverse applications such as video streaming/conferencing, voice telephony, and file download. Dynamic allocation of resources across all the users and all the applications provides an opportunity to achieve increasing network resource usage and to maximize the user satisfaction at the same time.

Application-driven cross-layer optimization (CLO) has been studied for systems supporting single applications [1–4]. However, in reality, the users sharing the wireless medium, for example, in a cell, usually run different applications. User satisfaction translates into a different set of requirements for each type of application. Furthermore, the impact of losses on the user-perceived quality is also very much application-dependent. Jointly optimizing the system for different users and applications requires (1) defining a common metric that quantifies the user perceived quality of

service for the service delivery and (2) mapping network and application parameters onto this metric.

The challenge of optimization across multiple applications has been treated mainly in the form of throughput maximization [5, 6]. Maximizing throughput leads to optimum performance only for applications which are insensitive to delay and packet loss. Multimedia applications such as video streaming and voice telephony are highly sensitive to changes in data rate, delay, and packet losses. Even the importance of a packet changes dynamically depending on the transmission history of previous packets. Due to these reasons, throughput maximization leads to performance which is usually not optimal with respect to user perceived quality for multimedia applications.

A possible metric to capture user satisfaction is the mean opinion score (MOS). MOS was originally proposed for voice quality assessment and provides a numerical measure of the quality of human speech at the destination. The scheme uses subjective tests (opinionated scores) that are mathematically averaged to obtain a quantitative indicator of the system performance. To determine MOS, a number of listeners rate the quality of test sentences read aloud over the communication circuit by a speaker. A listener gives each sentence a rating as follows: (1) bad; (2) poor; (3) fair; (4) good; (5) excellent. The MOS is the arithmetic mean of all the individual scores.

The multiapplication CLO approach proposed in this paper extends the use of MOS as a user-perceived quality metric to other applications, such as video streaming, web browsing and file download. This enables us to optimize across applications using a common optimization metric. The objective function to be maximized can be chosen, for example, to be the average MOS of all the users competing for the resources of the wireless communication system:

$$F(\tilde{\mathbf{x}}) = \frac{1}{K} \sum_{k=1}^K \lambda_k \cdot \text{MOS}_k(\tilde{\mathbf{x}}), \quad (1)$$

where  $F(\tilde{\mathbf{x}})$  is the objective function with the cross-layer parameter tuple.  $\tilde{\mathbf{x}} \in \tilde{X} \cdot \tilde{X}$  is the set of all possible parameter tuples abstracted from the protocol layers representing a set of candidate operation modes.  $\lambda_k$  are free parameters which can be chosen in two different ways. For a priority-based scheme, they can be chosen to provide different relative importance of the user as determined by the service agreement between the user and the service provider. For an equal-priority system,  $\lambda_k$  can be chosen to ensure fairness among the users. In this paper, we take the second approach. Although the MOS functions for different applications can be different, a linear combination, as in (1), can be used because the range of the functions is the same, that is, from 1 to 5. The decision of the optimizer can be expressed as

$$\tilde{\mathbf{x}}_{\text{opt}} = \arg \max_{\tilde{\mathbf{x}} \in \tilde{X}} F(\tilde{\mathbf{x}}), \quad (2)$$

where  $\tilde{\mathbf{x}}_{\text{opt}}$  is the parameter tuple which maximizes the objective function. Once the optimizer has selected the optimal values of the parameters, it distributes them to all the individual layers which are responsible for translating them back into actual layer-specific modes of operation.

In this work, the abstracted parameters for the physical and data link layers are transmission rate  $R$  and packet error probability (PEP) for all users for all candidate modes of operation. For a detailed description of the principle of parameter abstraction and the formulation of objective functions for multiuser cross-layer optimization, please refer to [1–3, 7].

The proposed MOS-based optimization approach has several advantages with respect to previous work. First, compared to traditional techniques for multiuser diversity [8], it allows us to directly relate network parameters, such as rate ( $R$ ) and packet error probability (PEP) to a user-perceived application quality metric such as MOS. Second, compared to the application-driven cross-layer optimization described in [2, 3], it allows us to further maximize the optimization gain by taking advantage of the diversity not only across multiple users running the same application, but also across users running different applications. Our experiments applied to scenarios including multiple concurrent video streaming, voice telephony, and file download applications show that MOS-based optimization significantly outperforms throughput-based optimization.

This paper is arranged as follows. In Section 2, we describe MOS functions for three different applications, namely voice telephony, file download, and video streaming.

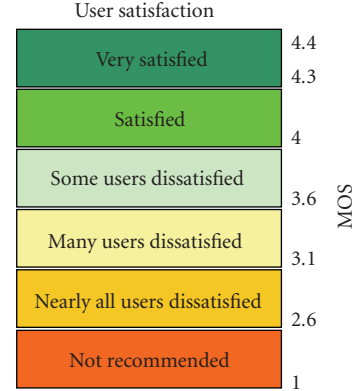


FIGURE 1: Relation between MOS and user satisfaction [9].

In Section 3, we give a detailed description of our multiapplication cross-layer optimization framework. Section 4 gives an overview of our simulation setup that is used to compare our approach with throughput maximization. Section 5 presents our experimental results and Section 6 concludes the paper.

## 2. MEAN OPINION SCORE (MOS)

The objective function of (1) requires the mapping of transmission characteristics (in our case transmission rate and packet error probability) to MOS for different applications. We now describe this mapping for voice communication, file download, and video streaming applications.

### 2.1. Voice communication

The traditional method of determining voice quality is to conduct subjective tests with panels of human listeners. The results of these tests are averaged to give MOS but such tests are expensive and are not feasible for online voice quality assessment. For this reason, the ITU-T has standardized a model, perceptual evaluation of speech quality (PESQ) [10], an algorithm that predicts with high correlation the quality scores that would be given in a typical subjective test. This is done by making an intrusive test and processing the test signals through PESQ.

PESQ measures one-way voice quality: a signal is injected into the system under test and the degraded output is compared by PESQ with the input (reference) signal. The output of the PESQ algorithm is a numerical value that corresponds to MOS. The mapping between MOS and user satisfaction is presented in Figure 1.

The PESQ algorithm is computationally too expensive to be used in real-time scenarios. To solve this problem, we propose a model to estimate MOS as a function of the transmission rate  $R$  and the packet error probability (PEP). The available rate determines the voice codec that can be used. In Figure 2 we show experimental curves for MOS estimation as a function of PEP for different voice codecs. The curves are drawn using an average over a large number of voice samples and channel realizations (packet loss patterns).

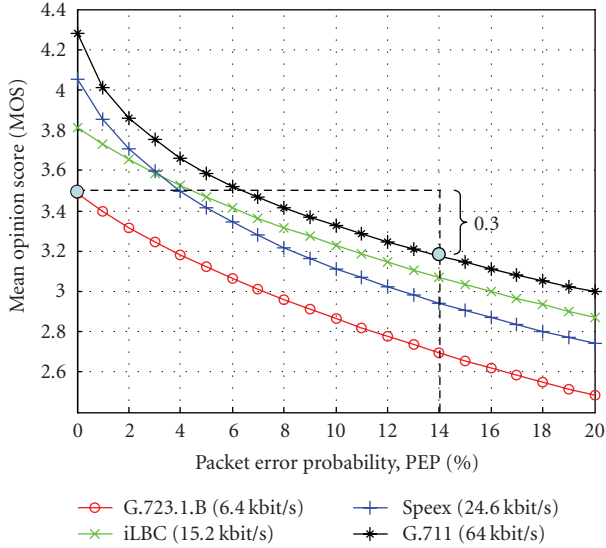


FIGURE 2: PESQ-based MOS versus packet error probabilities for different voice codecs.

These curves can be stored at the optimizer for every codec that is supported. If transcoding from an unsupported codec is required, the corresponding curve has to be signaled to the optimizer as side information. It should be noted that the average MOS (averaged over a large number of packet loss patterns for a fixed PEP) of individual voice samples may differ as much as 10% for the highest considered PEP, but the deviation from the mean values (averaged over a large number of voice samples) as shown in Figure 2 is found to be less than 7%.

Depending on the distortion imposed by the source codec, every voice codec leads to a different MOS value in the case of error-free transmission. Also the codecs exhibit different sensitivities to packet losses. As an example, let us consider two lower layer parameter tuples ( $R = 64$  kbps,  $PEP = 14\%$ ), and ( $R = 6.4$  kbps,  $PEP = 0\%$ ) and assume these two represent possible operating modes of the lower layers for a particular user. In this example, the second parameter tuple ( $R = 6.4$  kbps,  $PEP = 0\%$ ) leads to a gain of 0.3 on the MOS scale and the cross-layer optimizer would select it as its outcome.

## 2.2. File download

To estimate user satisfaction for file download applications, we use the logarithmic MOS-throughput relationship introduced in [11] which results from the assumption that the utility of an elastic traffic (e.g., FTP service) is an increasing, strictly concave, and continuously differentiable function of throughput. We assume that every user has subscribed for a given data rate and user satisfaction is characterized by the actual rate the user receives. The MOS is estimated based on the current rate  $R$  offered to the user by the system and packet

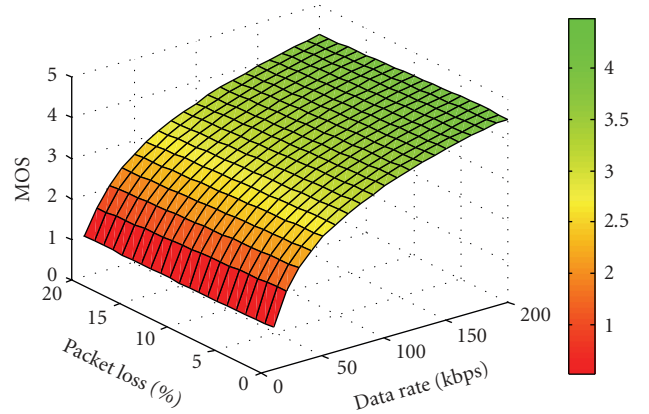


FIGURE 3: MOS as a function of transmission rate and packet error probability for file download applications.

error probability PEP:

$$MOS = a * \log_{10} [b * R * (1 - PEP)], \quad (3)$$

where  $a$  and  $b$  are determined from the maximum and minimum user perceived quality. If a user has subscribed for a specific rate  $R_{\text{service}}$  and receives  $R = R_{\text{service}}$ , then in case of no packet loss user satisfaction on the MOS scale should be maximum, that is, 4.5. On the other hand, we define a minimum transmission rate (e.g., 10 kbps in Figure 3) and assign to it a MOS value of 1. Using the parameters  $a$  and  $b$ , we fit the logarithmic curve in (3) for the estimated MOS. Varying the actual transmission rate  $R$  and packet error probability, PEP, this model results in the MOS surface shown in Figure 3.

## 2.3. Streaming video

Assessment of video quality is addressed in the literature with a wide variety of techniques. References [12, 13] are ITU recommendations to perform subjective assessment of TV and multimedia quality, respectively. Reference [14] gives a perceptual quality metric with respect to blockiness in compressed video. In [15], authors propose a reference-free method to estimate subjective quality using blurriness of the reconstructed video. Assuming that human visual perception is highly adapted for extracting structural information from a scene, [16] proposes a method of image quality assessment using degradation of structural information and develops a structural similarity index (SSIM). Reference [17] gives a comparison of different computational models of video quality, carried out by the video quality experts group (VQEG) of ITU.

Peak signal-to-noise ratio (PSNR) is an objective measurement of video quality which is widely used due to its simplicity and high degree of correlation with subjective quality [17]. PSNR is based on mean square error (MSE) as follows:

$$PSNR = 10 * \log_{10} \frac{255^2}{MSE}. \quad (4)$$

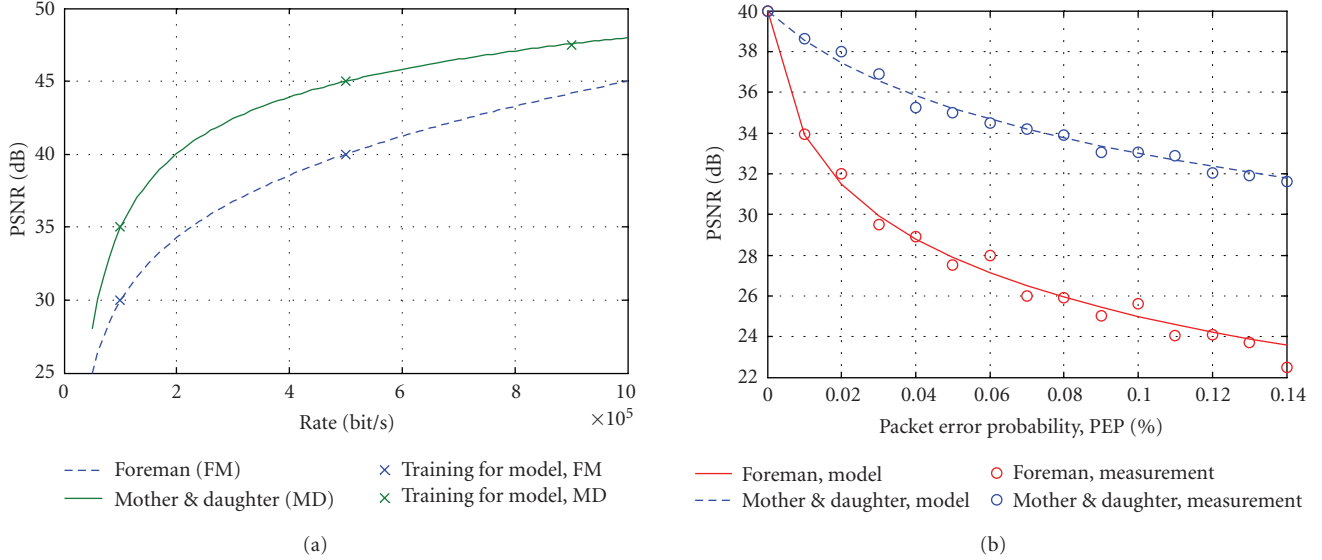


FIGURE 4: Illustration of the source distortion model (left), and the loss distortion model (right) using two test video sequences “Foreman” and “Mother and Daughter.”

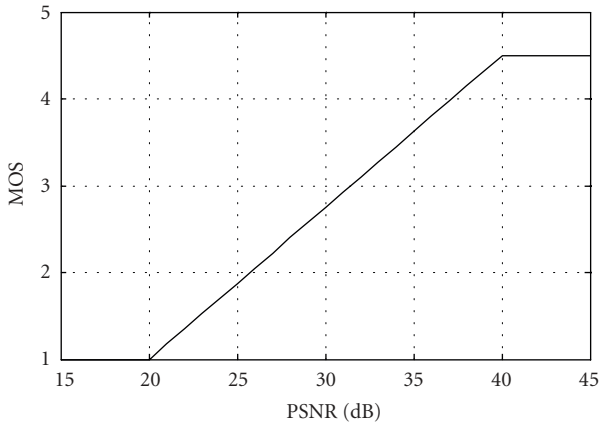


FIGURE 5: MOS versus PSNR.

The distortion of a video sequence can be expressed in terms of MSE. Distortion is assumed to be composed of two components, namely the *source* distortion  $D_S$  and the *loss* distortion  $D_L$ :

$$D = D_S + D_L, \quad (5)$$

$D_S$  is due to the compression of the video sequence, while  $D_L$  is due to the losses generated in the network. Consequently,  $D_S$  depends on the video source rate,  $R$  and  $D_L$  is a function of packet error probability PEP. In this paper, we apply the source distortion model as proposed in [18], and the loss distortion is assumed to be a linear function of PEP [19],

$$D = D_S + D_L = \frac{a}{\exp(R/b) - 1} + \beta \cdot \text{PEP}, \quad (6)$$

where  $a$ ,  $b$ , and  $\beta$  are model parameters. The source distortion model requires three pairs of rate and distortion measurements, as illustrated in Figure 4(a) for two test video

sequences. The loss distortion model requires measuring distortion for different PEP and uses best-fit to compute  $\beta$ .  $\beta$  is assumed to be independent of the video encoding rate. The validation of the loss distortion model is shown in Figure 4(b). Encoding is done with the H.264 reference encoder, with 30 frames per second in QCIF format. Each packet is assumed to have a fixed size of 125 bytes. Each video frame is encapsulated into one or more such packets.

In this work, we assume a simple linear mapping between PSNR and MOS. We assume that the maximum user satisfaction is achieved for a PSNR of 40 dB and the minimum user satisfaction results for PSNR values below 20 dB. The upper limit comes from the fact that reconstructed video sequences with 40 dB PSNR are almost indistinguishable from the original and below 20 dB very severe degradations distort the video. Figure 5 shows our assumed relationship between PSNR and MOS.

### 3. MULTIAPPLICATION CROSS-LAYER OPTIMIZATION

Based on the MOS framework described above, we are able to optimize the system taking actual user perceived quality of service into account. Our optimization scheme is not only applicable to the application types described in Section 2, but to any general mix of applications.

#### 3.1. Architecture

In [1, 2], we have proposed a cross-layer optimization architecture (Figure 6) with a component, called cross-layer optimizer (CLO), that periodically selects the optimal parameter settings of the different layers. This architecture is inspired by the CLO approach presented in [7]. Our CLO uses abstractions of different layers and optimizes the assignment of resources to each user. In our work, the abstracted parameters from the lower layers are rate  $R$  and packet er-

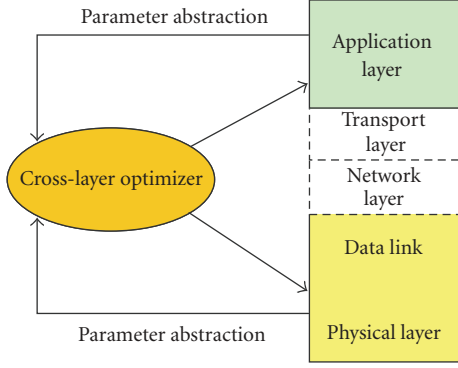


FIGURE 6: CLO architecture.

ror probability PEP for every user for all possible modes of operation. From the application layer we extract the utility functions (MOS versus PEP). We assume that the optimizer is located at the base station, the utility functions are generated at the sender and are sent as side information along with the media bitstream.

### 3.2. Optimization policy

As an example, we consider three types of users:  $U$ —requesting voice service,  $V$ —file download, and  $W$ —video streaming. Depending on the type of application, the mobile users require different resources over the wireless channel. The available transmission rate for each user depends on the modulation scheme, the channel code rate, and the assigned share of the medium access. In our example, a user requesting voice service may be served with different voice codecs (G.711, Speex, iLBC, or G.723.1.B), his data may be encoded with different channel code rates 1/2, 1/3, 1/4, or 1 (uncoded) and DBPSK or DQPSK modulation can be used. Every transmission policy gives different quality of service to the user and requires different amount of channel resources.

We create sets of transmission policies for every service.  $T_U$  is the set of transmission policies for voice service,  $T_V$  is the set of transmission policies for the file download service, and  $T_W$  is the set of transmission policies for the video service.

### 3.3. Mean opinion score maximization

The goal of this optimization is to achieve maximum user satisfaction and fairness among the users. For every user, depending on the service, we define a decision variable for every transmission policy—whether this user is served with a given transmission policy or not. Consequently, these decision variables are of boolean type, that is, either the user transmits its information using this policy or not. For the voice users, we have decision variables  $u_{ij}$ , where “ $i$ ” denotes the  $i$ th user and “ $j$ ” refers to the  $j$ th transmission policy available for the voice users.

Mobile users in the wireless network have time-varying position, which results in variable SNR at the receiver. Based on the SNR, we compute an estimate of the PEP [20] for dif-

ferent modulation schemes (DBPSK and DQPSK) and different channel code rates, that is, for all candidate transmission policies. A channel realization is generated and the estimation of the PEP is performed for all the transmission policies given the particular SNR at the receiver.

Our objective function for multiuser multiapplication cross-layer optimization is defined in (7). A maximization of the sum of the MOS perceived by every user in our multimedia wireless network has to be achieved. The parameter  $\lambda$  is used to ensure fairness among the users.

Maximize

$$\begin{aligned} & \sum_{i \in U} \sum_{j \in T_U} \lambda_{ui} u_{ij} E[\text{MOS}_{ij}] + \sum_{i \in V} \sum_{j \in T_V} \lambda_{vi} v_{ij} E[\text{MOS}_{ij}] \\ & + \sum_{i \in W} \sum_{j \in T_W} \lambda_{wi} w_{ij} E[\text{MOS}_{ij}] \end{aligned} \quad (7)$$

subject to

$$\begin{aligned} \sum_{j \in T_U} u_{ij} &= 1, \quad \forall i \in U, \\ \sum_{j \in T_V} v_{ij} &= 1, \quad \forall i \in V, \\ \sum_{j \in T_W} w_{ij} &= 1, \quad \forall i \in W, \end{aligned} \quad (8)$$

$$\begin{aligned} \sum_{i \in U} \sum_{j \in T_U} r_{ij} u_{ij} + \sum_{i \in V} \sum_{j \in T_V} r_{ij} v_{ij} + \sum_{i \in W} \sum_{j \in T_W} r_{ij} w_{ij} \\ \leq \text{total symbol rate.} \end{aligned} \quad (9)$$

In our example, every user must be associated with only one transmission rate, channel code rate, and modulation scheme. The decision variables  $u_{ij}$ ,  $v_{ij}$ , and  $w_{ij}$  are of boolean type which leads to the constraints (8). The total available symbol rate for all the users is constrained to be less than the total symbol rate of the system. Every transmission policy has an associated symbol rate  $r_{ij}$  and the sum of all the chosen symbol rates of all the users must be less than or equal to the total symbol rate. The above problem can be solved with a full search through the possible parameter space which has the worst case number of searches of  $|T_U|^{K_U} \cdot |T_V|^{K_V} \cdot |T_W|^{K_W}$  where  $|T_U|$ ,  $|T_V|$  and  $|T_W|$  are the numbers of transmission policies and  $K_U$ ,  $K_V$ ,  $K_W$  are the numbers of users of user classes  $U$ ,  $V$ , and  $W$ , respectively.

The parameters  $\lambda_{ui}$ ,  $\lambda_{vi}$ ,  $\lambda_{wi}$  in (7) are inserted to ensure a fair allocation of resources. The optimizer finds a resource allocation which maximizes the user satisfaction based on MOS. In this case, there is a possibility that even though the system performance is maximized, a given user is not satisfied. This could be caused by low receiver SNR and the optimizer can decide to allocate the resources to the other users. This contradicts with the fairness we are trying to offer to the users independent of their location. To solve this problem, we propose to select the scaling coefficients  $\lambda_{ui}$ ,  $\lambda_{vi}$ ,  $\lambda_{wi}$  based on the history of the user estimated MOS. On every rate allocation procedure, we find the user with the maximum average of the estimated MOS from the previous steps. Let us assume that we are at rate allocation step “ $N$ ” and we have  $K$  users in

the system. The value of the maximum perceived MOS by a single user is found by

$$\begin{aligned} \text{MaxMOS}_N \\ = \frac{1}{N-1} \max \left( \sum_{n=1}^{N-1} \text{MOS}_{1n}; \sum_{n=1}^{N-1} \text{MOS}_{2n}; \dots; \sum_{n=1}^{N-1} \text{MOS}_{Kn} \right). \end{aligned} \quad (10)$$

The scaling coefficient for every user is calculated with

$$\lambda_{kN} = \frac{\text{MaxMOS}_N}{(1/(N-1)) \sum_{n=1}^{N-1} \text{MOS}_{kn}}, \quad k = 1 \dots K. \quad (11)$$

The user with the maximum perceived MOS has a scaling coefficient of one. The other users have scaling coefficients in the range [1; 4.5], because the denominator is also bounded in the interval [1; MaxMOS<sub>N</sub>]. Since these  $\lambda$  values scale the estimated MOS for every transmission policy and we maximize the sum of the MOS of all the users, the optimizer assigns transmission policies with high estimated MOS to the users with higher  $\lambda$ . This ensures fairness by providing higher resources to the users which have been receiving lower MOS up to the time of the current optimization step.

### 3.4. Throughput maximization

A common network performance metric is the throughput of the system. Traditionally, the goal of the network operator is to maximize the network throughput. By throughput we consider the effective rate (goodput)  $G_{ij}$  of a given user  $i$  at time  $j$ :

$$G_{ij} = R_{ij} * (1 - \text{PEP}_{ij}) \quad (12)$$

with  $R_{ij}$  is the actual transmission rate and  $\text{PEP}_{ij}$  is the packet error probability. The objective function for such an optimization model is to maximize the sum of the goodput allocated to all the users in the system and is given with (13). The assumption is that a higher goodput will result in a higher user satisfaction regardless of the application type.

For throughput maximization, we have the same set of decision variables as in (7)–(9). The difference is the absence of the scaling parameter  $\lambda$ . Here we do not need scaling of the allocated transmission rate, because the transmission rates required by different applications are not comparable. Additionally, in order to make a fair comparison with our MOS-based optimization, we include a constraint on the packet error probability,  $\text{PEP}_{\max}$ , for each application type, so that the real-time applications are assigned a *sensible* share of the resources.

Maximize

$$\sum_{i \in U} \sum_{j \in T_U} u_{ij} G_{ij} + \sum_{i \in V} \sum_{j \in T_V} v_{ij} G_{ij} + \sum_{i \in W} \sum_{j \in T_W} w_{ij} G_{ij} \quad (13)$$

subject to

$$\sum_{j \in T_U} u_{ij} = 1, \quad \forall i \in U, \quad (14)$$

$$\sum_{j \in T_V} v_{ij} = 1, \quad \forall i \in V, \quad (15)$$

$$\sum_{j \in T_W} w_{ij} = 1, \quad \forall i \in W, \quad (16)$$

$$\sum_{i \in U} \sum_{j \in T_U} r_{ij} u_{ij} + \sum_{i \in V} \sum_{j \in T_V} r_{ij} v_{ij} + \sum_{i \in W} \sum_{j \in T_W} r_{ij} w_{ij} \quad (17)$$

$\leq$  total symbol rate,

$$\text{PEP}_i \leq \text{PEP}_{\max, i}. \quad (18)$$

### 3.5. Greedy resource allocation algorithm

The full-search resource allocation described in Sections 3.3 and 3.4 becomes computationally infeasible as the number of users in the system grows. For example, with three voice users, two ftp users and two video users, the number of resource allocations that have to be considered is  $4.845 \cdot 10^{12}$ .

The greedy allocation algorithm used in this work is similar to the work in [19]. It is initialized by assigning equal amount of resources to every user. In each subsequent step, a small amount of resources is taken from the user with the lowest sensitivity to a decrease of resources and assigned to the user that receives the maximum benefit. This is repeated until there is no further improvement in the objective function. The greedy algorithm for the MOS-maximization is described below. The throughput maximization is performed in a similar way.

Let  $\Theta_i$  denote the utility function, and  $\alpha_i$  the share of resource (symbol rate) of user  $i$ . Then,  $\Delta\Theta_i$  denotes the change of utility for user  $i$  due to a change of its resource share,  $\Delta\alpha_i$ , where  $\sum_{i=1}^K \alpha_i = 1$ , that is, the sum of resource share over all the users in the system equals unity. The greedy allocation can be expressed as an iterative maximization of the incremental utility values of two users  $i$  and  $j$ :

$$\max_{(i,j) \in \{1, \dots, K\}} \frac{\Delta\Theta_i}{\Delta\Theta_j}, \quad i \neq j, \quad (19)$$

where  $\Delta\Theta_i$  and  $\Delta\Theta_j$  are changes of utility due to an increase,  $\Delta\alpha_i$ , and decrease,  $\Delta\alpha_j$ , of resource share for user  $i$  and  $j$ , respectively, and  $K$  is the total number of users.

### 3.6. Generalization

Our cross-layer optimization scheme is not limited to a certain mix of only those application types we described in Section 2, but it is applicable to any general scenario. In Section 3.3, we propose the MOS-based optimization scheme where it is assumed that the application-layer side information (SI) is provided to the cross-layer optimizer in the form of MOS-functions. In a more general scenario, we need to consider the case when some of the streams provide SI and some do not. For this purpose, we classify the streams into two categories: SI and non-SI streams. Our strategy for these

two groups is going to be as follows: MOS-maximization for the SI streams, and throughput maximization for the non-SI streams. An initial resource allocation (e.g., symbol rate) among the two classes would be necessary. This can be assigned with the information on QoS requirement for each class. We regard this as a separate optimization issue. In our simulations, the resources (symbol rates) among the two classes are assigned in proportion to the number of streams belonging to each class. Let  $K_M$  and  $K_T$  be the number of SI and non-SI streams, respectively. Then, for the purpose of our simulation, optimization problems of Sections 3.3 and 3.4 are modified only in (9) and (17) to incorporate the symbol rate constraints of (20) and (21) respectively, as follows:

$$\sum_{i \in U} \sum_{j \in T_U} r_{ij} u_{ij} + \sum_{i \in V} \sum_{j \in T_V} r_{ij} v_{ij} + \sum_{i \in W} \sum_{j \in T_W} r_{ij} w_{ij} \leq \text{Total Symbol Rate} \cdot \frac{K_M}{K}, \quad (20)$$

$$\sum_{i \in U} \sum_{j \in T_U} r_{ij} u_{ij} + \sum_{i \in V} \sum_{j \in T_V} r_{ij} v_{ij} + \sum_{i \in W} \sum_{j \in T_W} r_{ij} w_{ij} \leq \text{Total Symbol Rate} \cdot \frac{K_T}{K}, \quad (21)$$

where  $K$  is the total number of users.

#### 4. SIMULATION

The simulations shown in this paper are performed with the following parameter settings. We assume a total of seven simultaneous users in the wireless network. Three voice users, one male and two female voices, are used. The voice samples are 60 seconds long. The voice signal comes from the backbone network encoded with G.711 voice codec at 64 kbps. In the base station, following the optimization output, the signal could be transcoded to 6.4 kbps with G.723.1 codec, 15.2 kbps with iLBC codec, 24.6 kbps with Speex, or it can be transmitted without transcoding at 64 kbps.

Two users perform a file download using FTP. Both of them have subscribed for a service with maximum offered transmission rate of 192 kbps.

Two users are using video streaming service. The video sequences used for our simulation are “foreman” and “mother and daughter,” encoded with the H.264 reference software encoder. The GOP structure is I-P-P-..., encoded at 30 frames per second in QCIF resolution ( $176 \times 144$  pixels).

The  $\lambda$  values in (7) are all initialized to 1 in our experiments. The total available symbol rate is constant and we have examined three different cases: 500 Ksymbol/s, 1000 Ksymbol/s, and 1500 Ksymbol/s. The supported modulation schemes are DBPSK and DQPSK. Channel code rates of 1/2, 1/3, 1/4, and 1 (uncoded) are supported, using convolutional code.

To reflect user mobility, the receiver SNR for every optimization step is drawn randomly for every user from a uniform distribution from 5 dB to 25 dB. The system is active for 60 seconds and we assume that the average channel characteristics remain constant for 1.2 second periods, which results in 50 optimization loops.  $PEP_{\max}$  is set to be 0.1, 0.2, and 0.3 for video, voice, and ftp services, respectively.

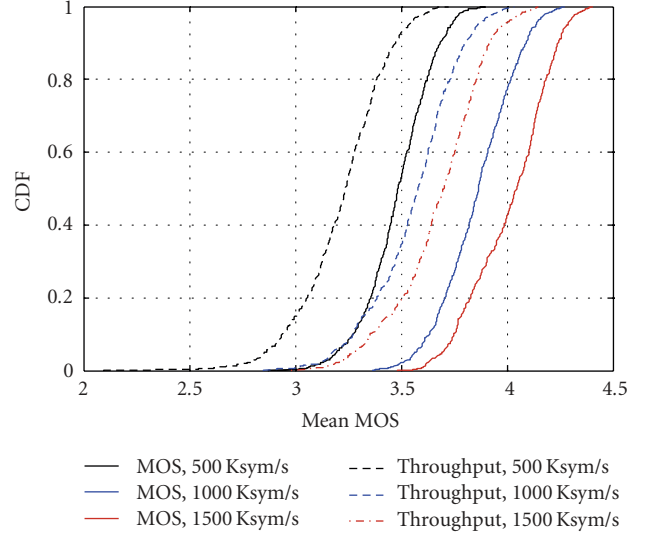


FIGURE 7: Mean opinion score of all seven users for three different total symbol rates (500, 1000, and 1500 Ksymbol/s) and two different optimization techniques, MOS maximization and throughput maximization.

The wireless system we have implemented in this work does not refer to any particular physical layer interface. We kept it intentionally simple, as the main goal of our work is to demonstrate the potential gain for any wireless system considering joint optimization across multiple different applications.

For the voice users, the signal samples are partitioned into 1.2 seconds and every sample is encoded using the voice codec determined by the optimization algorithm. At the end of the optimization loops, these voice samples are assembled into a single file and the perceived quality (MOS) is computed by comparing the original signal and the distorted one using PESQ.

For the video user, if a slice (packet) is lost, it is not written in the bit stream, which tells the decoder to invoke the error concealment algorithm. The PSNR of every frame and the resulting average PSNR are computed. The average PSNR is converted to an MOS value using the relationship shown in Figure 5. For file download we compute the MOS using the relationship given in (3).

## 5. RESULTS

### 5.1. Comparison between MOS-based and throughput-based optimization

In this section, a comparison between the two investigated optimization approaches (MOS maximization and throughput maximization) is performed. We use the setup described in the previous section and the results are based on 600 runs.

Figure 7 shows the cumulative density function (CDF) of mean opinion score over all the users for the two optimization approaches and three different total system rates: 500 Ksymbol/s (overloaded system), 1000 Ksymbol/s (moderately loaded system), and 1500 Ksymbol/s (lightly

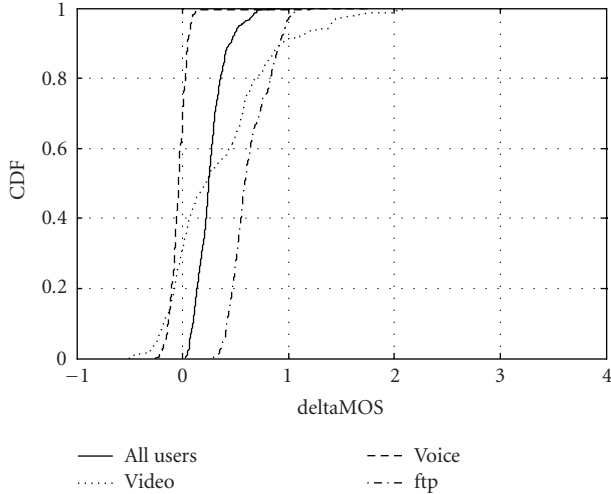


FIGURE 8: MOS gain per user, system symbol rate of 500 Ksymbol/s.

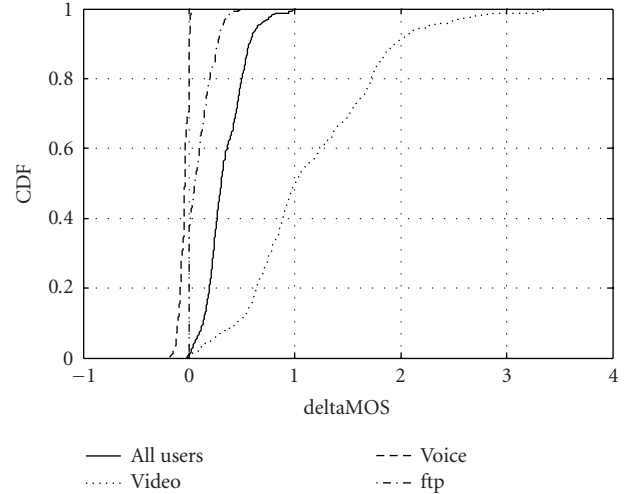


FIGURE 10: MOS gain per user, system symbol rate of 1500 Ksymbol/s.

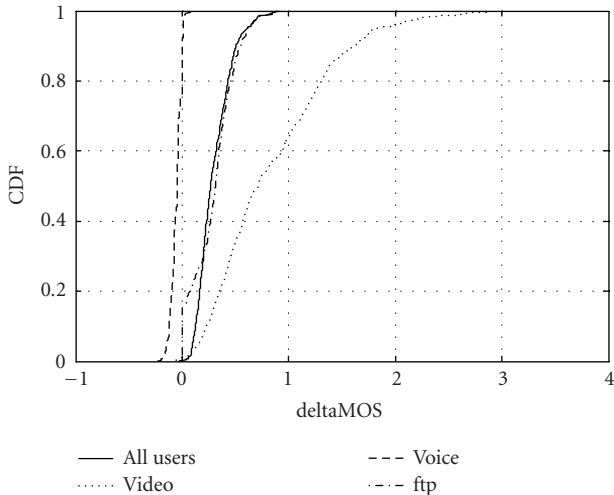


FIGURE 9: MOS gain per user, system symbol rate of 1000 Ksymbol/s.

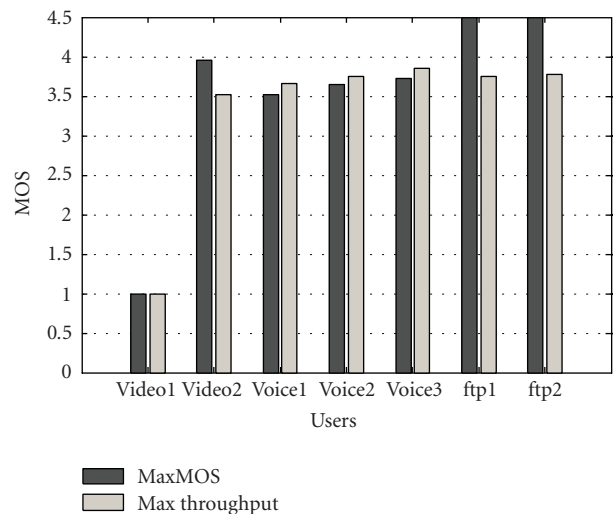


FIGURE 11: Bar plot showing the average MOS over a 30-second simulation run.

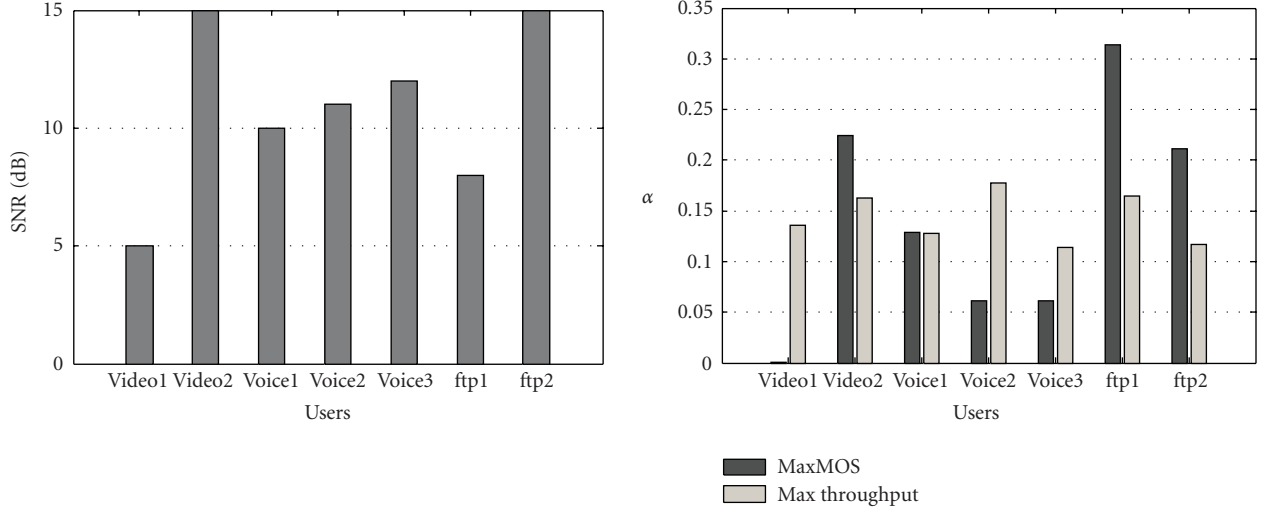
loaded system). The average gain in MOS for the three systems is 0.26, 0.30, and 0.34, respectively.

Figures 8–10 present the gain per user in the system. The curves are produced as a difference between the mean MOS computed with MOS maximization and throughput maximization. Starting with a system symbol rate of 500 Ksymbol/s (Figure 8), in 50% of the simulations, the average gain for all users is 0.26. The video and FTP users are benefited with a little penalty on the voice users. In Figures 9 and 10, we observe increasingly higher gain for the video users, with little noticeable loss of quality for the voice users. The quality of voice and video services are very sensitive to packet losses. In our throughput maximization approach we set a maximum allowable packet error probability,  $PEP_{\max}$  of 0.2 for voice and 0.1 for video service. This turns out to be a reasonable choice for the voice users and the second video user “(mother and daughter),” but is too high for the

first video user “(foreman),” because of its dynamic content which is highly sensitive to packet loss. In our MOS-based cross-layer optimization approach, application requirements are taken care of individually for each user, which results in optimum allocation of resources in terms of user perceived quality.

Figure 11 shows the average MOS of the seven users over a 30-second simulation run. Figure 12(a) shows the receiver SNR which was fixed during the simulation and Figure 12(b) shows the resource shares,  $\alpha$ . Video1 receives a very low SNR, which results in poor received video quality for both optimization approaches. However, our MOS maximization approach, being aware of the utility function of the applications, does not assign any resource to this user, and distributes the *saved* resources to other users which results in higher mean MOS.





(a) (b)  
FIGURE 12: (a) Mean receiver SNR of seven users, (b) resource shares.

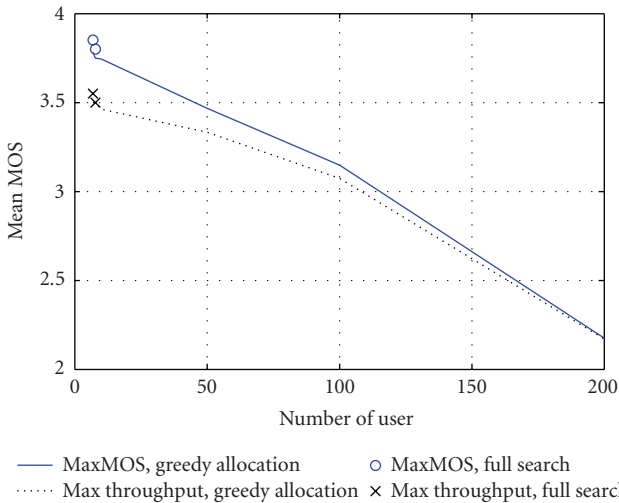


FIGURE 13: Mean MOS versus the number of users for the greedy algorithm and the full search algorithm.

## 5.2. Performance of the greedy search algorithm

The full-search algorithm described in Sections 3.3 and 3.4 is computationally too expensive to be implemented. In order to use our cross-layer optimization scheme in a real-time scenario, we also developed a greedy search algorithm, as described in Section 3.5. The resource that is allocated among the users is the time-share, which translates to a particular symbol-rate for a user, as the total symbol-rate of the system is fixed. Figure 13 shows the mean MOS versus the number of users using the greedy search approach for  $K = 7$  to 200 while the total symbol rate of the system is fixed at 1000 Ksymbol/s. In the simulations when we use varying number of user, we keep the number of video and ftp user fixed at two and two, respectively, and increase only the num-

ber of voice user. At  $K = 7$  and  $K = 8$ , we also compute the mean MOS using the full-search approach, and we observe little difference between the two approaches. For  $K > 8$ , the computation for the full-search approach becomes infeasible, while the greedy search remains fast enough to be used in on-line optimizations. As the number of user increases, the gap between the MOS-based and throughput-based approaches gradually decreases. Please note that for satisfied users, the MOS should stay above 3.5. At this level, we see significant improvements when using the MOS-based optimization.

The convergence speed of the greedy algorithm can be measured in terms of the number of iterations. The number of iterations tends to be dependent on the resource allocation step size  $\Delta\alpha_i$  and a minimum threshold of utility improvement at each iteration,  $\Delta\Theta_{th}$ . The improvement of utility at each iteration,  $\Delta\Theta_{incr} = \Delta\Theta_i - \Delta\Theta_j$  is compared with the threshold,  $\Delta\Theta_{th}$ . The algorithm is assumed to converge when  $\Delta\Theta_{incr} \leq \Delta\Theta_{th}$ . For a comparison of the number of iterations required for different number of users with a wide range of channel conditions, we keep these two parameters fixed,  $\Delta\alpha_i = 0.0001$  and  $\Delta\Theta_{th} = 0.00005$  (MOS).

Figure 14 shows the CDF of the number of iterations for 5, 10, and 50 users. The worst-case number of iterations is found to be in the range of 3000 to 4000 iterations. It is interesting to find that the 5-user case may take more iterations to converge than what we observe for the case of 50 users, the reason being the use of equal step size for both cases. Further fine-tuning is possible by choosing a step-size that is a function of the number of users. Also, for those applications, which have a limit on the rate (e.g., voice communication applications with at most 64 kbps), we can speed up the greedy algorithm by using this fact during initialization.

The time to complete each iteration, however, increases with the number of users. The convergence speed of the greedy algorithm in terms of time is shown in Figure 15. The measurements are taken from the Matlab-based simulation environment, with an Intel dual-core T2300 1.66 GHz

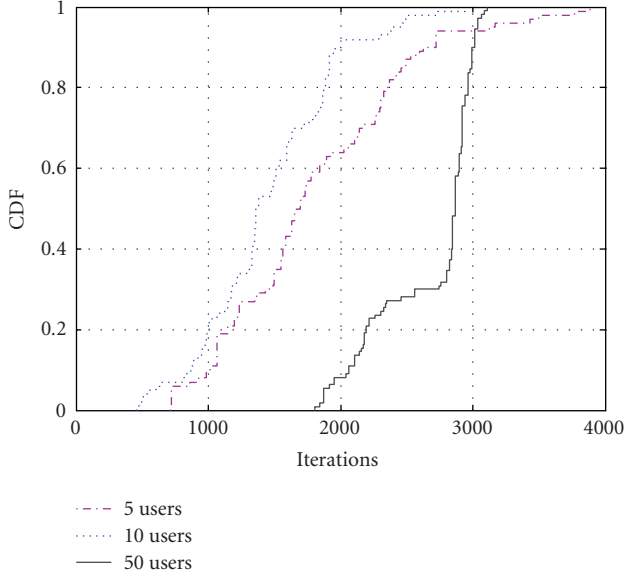


FIGURE 14: CDF of number of iterations with greedy algorithm.

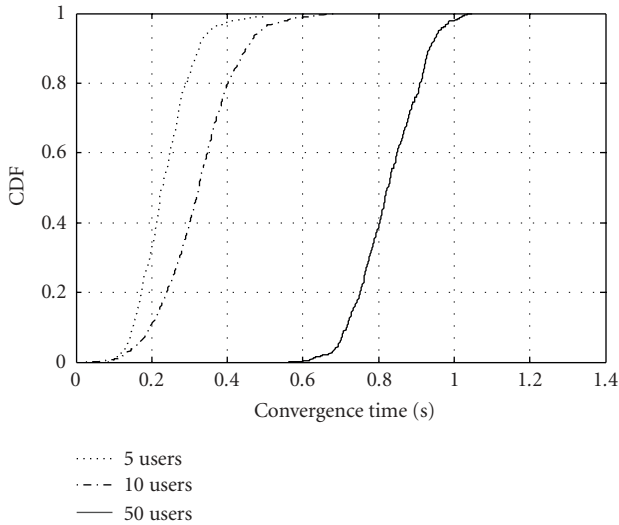


FIGURE 15: CDF of convergence time with greedy algorithm.

processor. Using dedicated software and hardware environments, the convergence speed is expected to be much faster.

Figure 16 shows the worst-case performance gap between full-search and greedy algorithm. The performance gap is computed as the difference between the MOS values obtained by using the full-search and the greedy algorithm. We find that the gap is reasonable.

### 5.3. Optimization with and without side information

In this section, we consider the more general case when some of the streams provide application-layer side information (MOS functions) and some do not. As discussed in Section 3.6, we perform MOS-based optimization for the SI streams, and throughput-based optimization for the non-SI

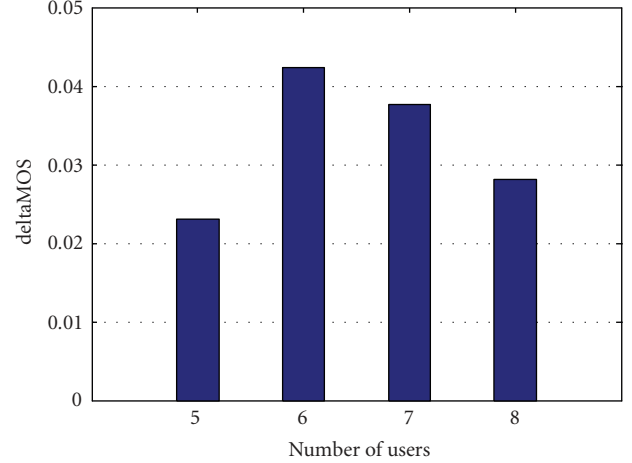


FIGURE 16: Worst-case difference in MOS (deltaMOS) between full-search and greedy algorithm.

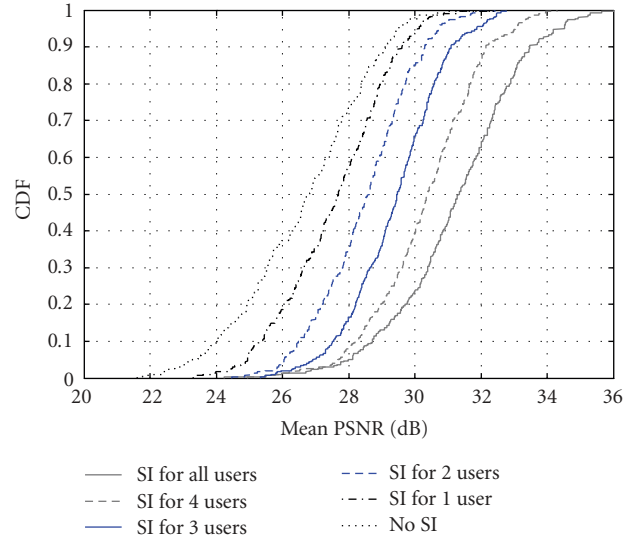


FIGURE 17: CDF of mean PSNR (in dB) for a five-user video streaming scenario.

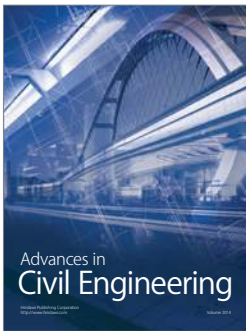
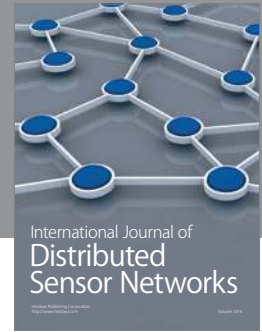
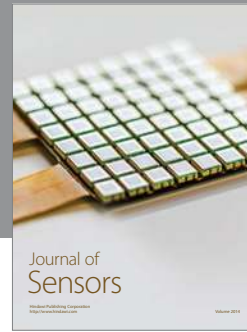
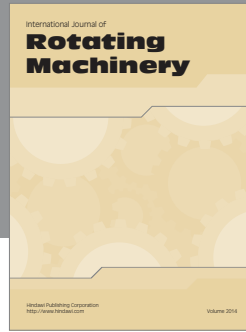
streams. In this section, we consider a five-user video streaming scenario. Figure 17 shows the CDF of mean PSNR over five users for all possible cases of with and without SI. For this scenario, we use the same video sequence (“foreman” sequence) for all five users with a wide variety of channel conditions. For the case “SI for all users” MOS-based optimization is used, while for the case of “no SI,” throughput-based optimization is performed. For the other cases, both approaches are used in combination. Figure 17 shows that we have an average gain of 1 dB PSNR for each additional stream with SI. It is easy to extend this strategy to a system having different application types, although the results will be more involved due to the different quality metrics for different applications.

## 6. CONCLUSION

In this paper, we propose a novel multiuser cross-layer optimization approach across multiple applications using MOS as a common application layer performance metric. With this approach we are able to dynamically optimize the wireless transmission system resource usage and the user perceived quality of service in a multiuser environment. We compare our approach to a traditional approach where allocation is done with the goal of maximizing overall throughput. Our simulation results show significant improvements in terms of user perceived quality for a variety of circumstances.

## REFERENCES

- [1] W. Kellerer, L.-U. Choi, and E. Steinbach, "Cross-layer adaptation for optimized B3G service provisioning," in *Proceedings of the 6th International Symposium on Wireless Personal Multimedia Communications (WPMC '03)*, pp. 57–61, Yokosuka, Japan, October 2003.
- [2] L.-U. 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, October 2004.
- [3] 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.
- [4] 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.
- [5] V. Tsibonis, L. Georgiadis, and L. Tassioulas, "Exploiting wireless channel state information for throughput maximization," in *Proceedings of the 22nd IEEE Annual Joint Conference of Computer and Communications Societies (INFOCOM '03)*, vol. 1, pp. 301–310, San Francisco, Calif, USA, March-April 2003.
- [6] X. Liu, E. K. P. Chong, and N. B. Shroff, "Transmission scheduling for efficient wireless utilization," in *Proceedings of the 20th IEEE Annual Joint Conference of Computer and Communications Societies (INFOCOM '01)*, vol. 2, pp. 776–785, Anchorage, Alaska, USA, April 2001.
- [7] M. T. Ivrlač and J. A. Nossek, "Cross layer optimization—an equivalence class approach," in *Proceedings of the International Symposium on Signals, Systems, and Electronics (ISSSE '04)*, Linz, Austria, August 2004.
- [8] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74–80, 2003.
- [9] ITU-T G.107, "The E-model, a computational model for use in transmission planning".
- [10] ITU-T P.862, "PESQ: an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs".
- [11] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, 1997.
- [12] ITU-R Recommendation BT.500, "Methodology for the subjective assessment of the quality of television pictures".
- [13] ITU-T Recommendation P.910, "Subjective video quality assessment methods for multimedia applications".
- [14] S. Winkler, A. Sharma, and D. McNally, "Perceptual video quality and blockiness metrics for multimedia streaming applications," in *Proceedings of the 4th International Symposium on Wireless Personal Multimedia Communications (WPMC '01)*, pp. 547–552, Aalborg, Denmark, September 2001.
- [15] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "A no-reference perceptual blur metric," in *Proceedings of IEEE International Conference on Image Processing (ICIP '02)*, vol. 3, pp. 57–60, Rochester, NY, USA, September 2002.
- [16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [17] Final report from the Video Quality Experts Group, "On the validation of objective models of video quality assessment," October 2003.
- [18] L. U. Choi, M. T. Ivrlač, E. Steinbach, and J. A. Nossek, "Sequence-level models for distortion-rate behaviour of compressed video," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 2, pp. 486–489, Genova, Italy, September 2005.
- [19] D. Jurca and P. Frossard, "Media-specific rate allocation in multipath networks," *IEEE Transactions on Multimedia*, vol. 9, no. 5, 2007.
- [20] 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.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

