# Quality of Experience based Optimization of Heterogeneous Multimedia Sessions in IMS

Agata Brajdic
University of Zagreb
Faculty of EE and Computing,
Unska 3, 10000 Zagreb, Croatia
agata.brajdic@fer.hr

Andreas Kassler
Karlstad University
CS Dept., Universitetsgatan 2,
65188 Karlstad, Sweden
andreas.kassler@kau.se

Maja Matijasevic
University of Zagreb
Faculty of EE and Computing,
Unska 3, 10000 Zagreb, Croatia
maja.matijasevic@fer.hr

***Abstract*—For delivering multimedia services over (wireless) networks it is important that the mechanisms which negotiate and optimize content delivery under resource constraints take into account user perceived quality in order to improve user satisfaction. Within the scope of the IP Multimedia Subsystem (IMS) architecture a novel application server may be added which handles multi-user multi-flow Quality of Experience (QoE) negotiation and adaptation for heterogeneous user sessions. Based on a mathematical model, which takes into account the characteristics of audio, video and data sessions for QoE optimization, we develop several optimization algorithms to be used by the application server to maximize overall user defined QoE parameters for all ongoing multimedia sessions, subject to network resource constraints. Our results show that a greedy based approach provides a reasonable compromise in terms of run-time and sub-optimality for the overall QoE based resource allocations.**

***Index Terms*—multimedia communication, communication systems, resource management, algorithm, IMS**

## I. Introduction

In current networked multimedia systems, dynamic resource management and negotiation are based on measurable quantitative parameters such as bandwidth, delay, jitter, and loss ratio, usually referred to as Quality of Service (QoS). Recent trends in designing and evaluating multimedia systems, however, are becoming more user-centered, because the user ultimately judges the quality he experiences. The impact of resource limitations on user experience is highly related to media type and codec involved. For example, losing an I-frame in an H.263 video stream has more severe implications than losing a B-frame, or, reducing the data-rate of a HDTV video stream by 40 kbps has less impact than reducing its audio flow by the same amount. This paradigm shift leads to a system design focusing on the Quality of Experience (QoE) as perceived by the user of a given service. As a consequence, for dynamic negotiation and allocation of resources across all users and all sessions, the ultimate goal for a network service provider should be to control network resource usage while at the same time maximizing the QoE. Such optimization is difficult to achieve as (possibly many) diverse applications, and corresponding multimedia sessions, have to be supported concurrently with varying levels of resource demands. A resource optimization for a variety of users and sessions requires metrics that quantify the QoE of a user for a session, and a mapping process between network and application parameters onto those metrics. Further more, a negotiation process is needed to take into account limited resources and willingness of users to accept reduced QoE in case of resource shortage. Finally, the process of deciding which media flows should operate at what resource level in order to maximize the user perceived QoE translates into a constrained optimization process.

We address these issues within the scope of the IP Multimedia Subsystem (IMS) [1]. This paper builds upon the idea of deploying a QoS Matching and Optimization Application Server (QMO AS) in the IMS, presented in more detail in [2] [3]. The QMO AS provides advanced parameter matching and optimization procedures throughout the QoS negotiation process. QMO AS uses SIP over standard IMS interfaces and introduces additional knowledge about the user and service by using respective XML-based “profiles”. While the initial QMO AS allows an adaptation of a single user session composed of multiple media flows, it does not consider optimizing resources among multiple sessions, and its operation is limited to “traditional” QoS parameters without considering QoE. In this paper, we extend the functionality of the QMO AS towards QoE based negotiation and optimization, while taking into account multiple concurrent sessions, belonging to various (types of) multimedia applications. We develop a mathematical model that characterizes overall user satisfaction under resource constraints in order to yield a set of configurations of multiple user sessions which maximizes the overall user perception of service quality across all domain users. This is achieved by prioritizing users and services based on contracts between users and providers, user preferences and budget limitations. As such constraint optimization problem is hard to solve, we develop, implement and test two heuristic optimization algorithms which allow the calculation of near optimal service configurations across multiple domain sessions with limited CPU requirements.

After a brief overview of related work in Section II, Section III summarizes the proposed extended QMO AS functionality. Section IV presents the mathematical model, and Section V describes the proposed algorithms. Section VI presents the evaluation of the proposed algorithms illustrated by an example. Section VII concludes the paper.

## II. Related Work

Application-driven multimedia resource optimization found in literature mainly focuses on networks having a single service type (e.g., VoIP or video delivery) [4]. In real systems, multiple users running diverse services coexist, leading to a diverse set of requirements to be optimized regarding user perceived quality. As the impact of resource constraints on service quality is service-dependent, optimizing a service towards limited resources has been tackled mainly in the form of throughput maximization (e.g. [5]). This leads to suboptimal performance for applications sensitive to delay and loss, such as live video conferencing and IP telephony, and consequently, to suboptimal user perceived quality.

Regarding metrics for QoE management, Mean Opinion Score (MOS) has been used in many works to characterize user satisfaction for different types of streamed media content, mostly audio and video. The MOS related utility functions adopted for the purposes of this work are described next.

### *A. Streaming Video MOS related Utility Function*

We use video distortion as an input to derive user utility for streaming video [6]–[8]. Several proposals address the assessment of video quality such as peak signal-to-noise ratio (PSNR) parameter, which is typically used to model video distortion function and is based on mean square error (MSE). PSNR is typically used to provide objective measurement of video quality because it is simple to calculate and correlates nicely with subjective quality [9]. In the model adopted from [10], the end-to-end video distortion $D$ is composed of distortion at the source caused by video coding, $D_S$, and the channel distortion caused by packet loss, $D_L$, and calculated as follows:

$$D(R,\pi) = D_S + D_L = \alpha * R^{\xi} + \beta * \pi \tag{1}$$

where $R$ denotes the data rate of the video codec, $\pi$ the packet error probability, and $\alpha$, $\beta \in \mathbb{R}^+$ and $\xi \in [-1,0]$ are model parameters as defined by the properties of the encoder and video material. In contrast to [10], we do not assume a single QoS class to be used for the streamed video but allow multiple QoS classes. Therefore, target data rates and packet error probabilities can be defined for various QoS classes, for example, through a service configuration using a service profile.

### *B. Calculation of audio MOS*

The MOS value for the audio model can be calculated according to the R-factor [11], where MOS is expressed as:

$$\begin{aligned} MOS &= 1 + 0.035R \\ &+ 7 * 10 - 6R(R-60)(100-R) \qquad (2) \\ R &= 94.2 - I_e - I_d \qquad (3) \end{aligned}$$

Here, the parameter $I_d$ denotes the end-to-end delay impact on user quality perception, while the parameter $I_e$ reflects the impact of the packet loss on perceived quality. $I_d$ and $I_e$ can be estimated by using fitting mechanisms.

### *C. Calculation of data MOS value*

We use MOS based function to evaluate the quality of data session based on a FTP download model, according to [12]:

$$MOS = a * log[b * R * (1 - PEP)] \tag{4}$$

where $a$ and $b$ are again model parameters, $R$ is the flow rate, and $PEP$ parameter represents the packet loss ratio.

The design of our extended QMO AS is based on MOS mapping functions described above, and the development of better QoE metrics and mapping functions is out of scope of this article. The proposed approach, however, is generic enough that any applicable utility functions could be used, including those for other media types, and other metrics. The works [9], [13]–[15] take into account different generic network and media specific parameters and yield MOS estimation formulas based on the assessment provided by the set of independent human testers. Various QoE metrics have been proposed for images [16]. A summary of objective perceptual video quality metrics standardization may be found in [17].

## III. Overview of QMEX AS Functionality

The main goal of the QMEX AS (Quality of Multimedia Experience Application Server) is to assist the network operator in the optimization process in the network. In contrast to standard QoS based resource management, QMEX AS tries to balance resources among different and heterogeneous user multimedia flows so as to maximize the *overall* user experienced quality, subject to available resources in the domain. It thus retains a global view of the system resources and their current consumption, and performs global optimization of all currently active sessions within the domain based on a unified QoE metric. It also assists in negotiating user perceived quality levels for audio, video and data sessions within a domain while taking into account resource constraints, network utilization, and operator policies. This includes QoE and QoS parameter matching and optimization algorithms during session establishment, as well as during the course of a session, if needed. The goal of the negotiation process is to determine a set of feasible service parameters and select those which optimize a given objective function under given resource constraints. As an example, we develop an optimization function which allows to maximize the global system utility expressing the level of satisfaction of all users with provided service levels within the domain. As the system design is flexible, different objective functions can be supported such as to maximize the fairness among competing flows.

### *A. QMEX AS processes to determine final service profile*

A high-level view of the processes by which QMEX AS determines the final service profile is illustrated in Figure 1. During the session initiation process, the QMEX AS receives the *Client Profile* which is composed of the constraints imposed by a given user terminal, the characteristics of user's access and the user's personal preferences. For each session to be established, the Client Profile is matched with the corresponding *Service Profile*, describing requirements of

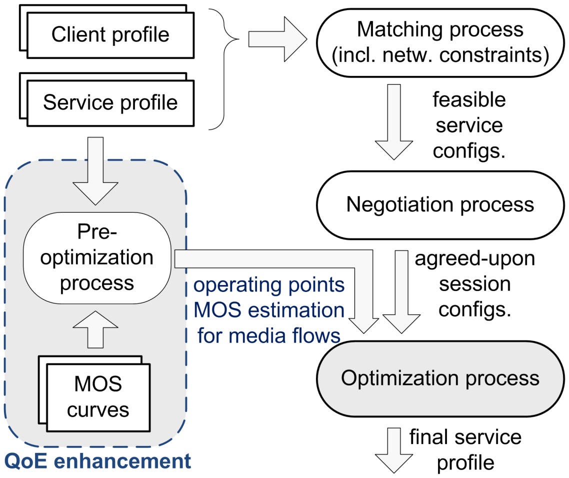
Client profile
Service profile
Matching process (incl. netw. constraints)
feasible service configs.
Negotiation process
agreed-upon session configs.
Pre-optimization process
operating points MOS estimation for media flows
MOS curves
Optimization process
final service profile
QoE enhancement

Fig. 1. Overview of the QMEX AS processes

the requested service, and with constraints imposed by the network operator (e.g., operator policy, resource limitation). This matching process yields a set of feasible service configurations. The negotiation process enables the end points to agree upon a subset of common service configurations supported by both end points and the network operator. Without the QoE enhancement (as described in [3]), the set of agreed-upon session configurations led to the optimization process, which generated the final service profile. With the proposed QoE-based enhancement, the pre-optimization process derives the utility values corresponding to different service configurations based on utility functions that allow to express user perceived quality (shown as "MOS curves"). This approach relieves the service provider from knowing the effect of a particular service configuration on the user perception of service quality under given network conditions. The set of negotiated service configurations mapped with utility values calculated during the pre-optimization process serves as an input into the optimization process that tries to maximize the degree of satisfaction of all users within the domain. Here, QMEX AS calculates the optimal utilization of network resources which would maximize user perceived quality under given network conditions. Furthermore, the optimization process takes into account restrictions imposed upon single user session by client equipment, access and preferences as well as network operator policies and service provider requirements affecting multiple sessions. Finally, the session establishment finishes with resource allocation procedures. An important aspect of the QMEX AS is the usage of the MOS as a measure for user perceived quality metric, which is applied in a unified way to audio, video and data download sessions. Such unified MOS metric allows the QMEX AS to easily optimize across multiple applications, using, for example, the average MOS, or overall MOS, as the optimization criterion.

### B. QoE based Utility Functions

In our system model, each media flow within a session is represented with the set of operating points that is passed to a global optimization engine as input. The operating point indicates the amount of resources required to attain a certain value of utility. The result of the optimization process will be for each user session and flow a given operating point, which in combination yields the maximum value of the system utility under given resource constraints. The utility assigned to an operating point indicates the degree of user satisfaction with the provided service, and thus, the level of achieved QoE. We leverage current research to evaluate the impact of network and coding schemes on user perceived quality by focusing on an overall MOS value as a sole optimization criterion. The MOS parameter by definition provides a numerical indication of the subjective user quality perception of received media on the scale from 1 to 5, where 1 stands for the lowest perceived quality, and 5 represents the highest one.

The service provider sends additional specific information to the QMEX AS about the media content that is offered along with a utility function per media flow to be used which allows the QMEX AS to calculate MOS related utility of different flow operating points in the pre-optimization process. This can be implemented using e.g. SIP with SDP extensions for the utility function. The utility function should allow a mapping of system resource availability indicated through parameters such as data rate, packet loss, and delay, to the flow's QoE as received by the user. The utility functions are later used by the QMEX AS to dynamically maximize QoE across all users and flows, subject to constraints imposed by various parties (e.g., network operator, service provider, or user).

To be able to calculate the impact on arbitrary resource constraints such as allocated bandwidth, packet loss rate, etc., on user perceived quality, the QMEX AS will be given a set of rate-loss-distortion mappings during a session setup at which distortion of the video content has been measured off-line using different parameter settings for packet loss and bandwidth. The rationale is that the distortion of video content comprised of fast moving scenes is under greater affect of packet loss than video composed of static scenes. Therefore, if a packet will be dropped, the user utility will be impacted in a different way depending on the motion within the scenes. Here, we refer an interested reader to [12], [9] for more detail. The QMEX AS can use curve fitting techniques to derive and store missing operational points. The content provider is thus able to select for the video to be transmitted such rate-loss-distortion mapping, which depends on the type of video content. These mappings are then signaled to the QMEX AS to locally store the definition of those parameterized distortion functions. Given those model parameters, the MSE distortion can be obtained for every rate/loss combination.

## IV. Formulation of Global Objective Function

The QMEX AS uses a mathematical model for optimization of multiple user supplied utility functions based on MOS as defined above which relates between a user (session) $u$, the respective media flow $i$, and a given operating point $j$. A given operating point can be associated with the level of resources required to satisfy the media flow. For example, a video flow can be encoded at 3 different visual quality levels (operating points) leading to three streams with different bandwidth requirements and possibly different resilience to errors. For each such operating point, impact of packet loss may be different but can be determined beforehand, and its utility value calculated as described above.

Let $U$ denote the set of active user sessions. Each session may consist of one or more media flows (audio, video, data). As the contents and the directions of flows need not be identical for all flows of the same type, the discrete set of operating points associated with each flow may vary. A resource vector $\mathbf{r_{uij}} = (r_{uij1}, ..., r_{uijq}, ..., r_{uijQ+C+K})$ describes resource consumption at a particular operating point needed to achieve a certain utility level. Each operating point of a given flow within the system, which belongs to a particular user session, is assigned to exactly one resource vector. The first $Q$ elements of this vector correspond to the bandwidth allocated in each of the $Q$ network QoS classes. Our model assumes that at any given point in time, one media flow content can not be simultaneously streamed in more than one QoS class, and thus, only one of the first $Q$ elements of resource vector can be greater than zero. The next $C$ elements jointly represent consumption of a single user resource (e.g., CPU, memory, budget), while the remaining $K$ elements pertain to resources of the network operator (e.g., a total number of flows within one QoS class) or the service provider (e.g., a total number of users being simultaneously served).

Each QoS network class is characterized with delay, jitter and loss parameters as well as the total bandwidth that the network provider has given to the service provider within a particular class. The service provider may offer the user to transmit his flow traffic with higher rates and higher loss or lower rates and lower loss for the same price. Which of these two options suits better to a particular user is given by the user's utility function. This concept will cover the scenario in which due to the increase of the network loss, a service may continue to be served within a different QoS class (yielding lower utility). Furthermore, it is assumed that the matching process performed earlier has eliminated those QoS classes and corresponding utility curves that are not supported either by the user or the service itself (e.g., operating point corresponding to HDTV 1080p video streaming to a mobile phone with small display can be eliminated directly).

The objective function used by QMEX AS (although any other function can be used) is to maximize the sum of all user utilities as:

$$\max \sum_{u \in U} \sum_{i=1}^{n_u} \sum_{j=1}^{p_{ui}} W_u w_{ui} x_{uij} MOS(\mathbf{r_{uij}}) \tag{5}$$

where $W_u$ denotes the user priority, $w_{ui}$ denotes media flow weight factor, $x_{uij}$ is a binary variable, and $MOS(\mathbf{r_{uij}})$ represents the MOS value at an operating point corresponding to the resource vector $\mathbf{r_{uij}}$.

The following constraint applies to all flow weight factors $w_{ui}$ of a particular user $u$:

$$\sum_{i=1}^{n_u} w_{ui} = 1, \ \forall u \in U \tag{6}$$

The factor $W_u$ represents the relative priority of the user $u$ for the particular service (e.g., a "premium" user requesting "bronze" VoIP service) over other users in the contention for the same network resources. Therefore, the factor $W_u$ serves as the means for differentiating user groups based on the specific criteria and is typically dependent on the type of application the user is being served with (e.g., streaming in a video conference may have higher priority over streaming of a video clip, even though both applications consist of audio and video flows), minimal accepted quality level of service configuration (typically labeled as HIGH, MEDIUM, and LOW), and the user's contract with the network provider. A weight factor allows a network provider to prioritize among different users and services in the optimization process. Setting a higher weight value would translate into having a greater impact in the overall optimization process. Furthermore, $x_{uij}$ is a binary variable utilized for the purpose of selecting exactly one flow operating point with the following constraint:

$$\sum_{i=1}^{n_u} \sum_{j=1}^{p_{ui}} x_{uij} = 1, \ \forall u \in U, \ x_{uij} \in \{0, 1\} \tag{7}$$

Next, we identify a set of constraints imposed by all involved parties (user, service provider, and network operator). The following constraints limit the total consumption of network resources (downlink and uplink bandwidth) across all users:

$$\sum_{u \in U} \sum_{i=1}^{h_u} \sum_{j=1}^{p_{ui}} \sum_{q \in Q} x_{uij} r_{uijq} \leq B^{dn} \tag{8}$$

$$\sum_{u \in U} \sum_{i=h_u+1}^{n_u} \sum_{j=1}^{p_{ui}} \sum_{q \in Q} x_{uij} r_{uijq} \leq B^{up} \tag{9}$$

where $B^{dn}$ and $B^{up}$ denote maximum amount of downlink and uplink bandwidth respectively, and $h_u$ stands for the number of flows pertaining to the user $u$ in the downlink direction. Such additional constraint has been introduced addressing the limit of the maximum bandwidth that a network operator may assign to a particular service provider in each of the QoS classes. We introduce sets $\{B_{v,1}^{uplink}, ..., B_{v,Q}^{uplink}\}$ and $\{B_{v,1}^{downlink}, ..., B_{v,Q}^{downlink}\}$ to denote a maximum network bandwidth assigned to a particular service provider $v$ in each of the QoS classes in uplink and downlink direction, respectively. The following constraints apply within a single QoS class:

$$\sum_{u \in U} \sum_{i=1}^{h_u} \sum_{j=1}^{p_{ui}} f(v,i) x_{uij} r_{uijq} \leq B_{v,q}^{downlink}, \tag{10}$$

$$\sum_{u \in U} \sum_{i=h_u+1}^{n_u} \sum_{j=1}^{p_{ui}} f(v,i) x_{uij} r_{uijq} \leq B_{v,q}^{uplink}, \tag{11}$$

$$\forall q \in Q, \forall v \in V$$

The set $V$ denotes the set of all active service providers having a contract with the network provider. The function $f(v,i)$ returns a binary value, which indicates whether the given flow $i$ (which belongs to user $u$) is being transmitted from/to the given service provider $v$. If it is transmitted, the function will return 1; otherwise it will return 0. Using this formulation, the scenario in which the user is simultaneously streaming content from different service providers can be addressed as well.

Aside from bandwidth restrictions, the network operator and service provider may impose additional constraints:

$$\sum_{u \in U} \sum_{i=1}^{n_u} \sum_{j=1}^{p_{ui}} x_{uij} r_{uijq} \leq R_q^{max}, \qquad \forall q \in (Q+C+1, ..., Q+C+K) \tag{12}$$

where the vector $R = (R_{Q+C+1}^{max}, ..., R_{Q+C+K}^{max})$ represents constrained resources. In order to introduce fairness into the model, we determine a minimum utility value per flow that needs to be satisfied. This minimum may be defined per user basis by specifying a vector of minimum utility values corresponding to each user, or per flow basis. When using the per user defined minimum, the additional constraint is:

$$\sum_{i=1}^{n_u} \sum_{j=1}^{p_{ui}} x_{uij} MOS(\mathbf{r_{uij}}) \geq MOS_u^{min}, \ \forall u \in U \tag{13}$$

The following constraint takes into account limited user capabilities:

$$\sum_{i=1}^{n_u} \sum_{j=1}^{p_{ui}} x_{uij} r_{uijq} \leq c_{uq}^{max}, \qquad \forall u \in U, \forall q \in (Q+1, ..., Q+C) \tag{14}$$

where capabilities vector $c_u = (c_{u,Q+1}^{max}, c_{u,Q+2}^{max}, ..., c_{u,Q+C}^{max})$ represents a set of capabilities pertaining to the user $u$. For the sake of simplicity, the elements of $c_u$ correspond to the elements of the resource vector at a given operating point. When using the limitations per flow (if such exist, e.g., max. rate per video flow), they may be introduced as follows:

$$\sum_{j=1}^{p_{ui}} x_{uij} r_{uijq} \leq R_{uiq}^{max}, \qquad \forall u \in U, \forall q \in 1, ..., K, \forall i \in 1, ..., n^u \tag{15}$$

## V. Iterative MOS Increase and Adapted Greedy algorithms

Since the above constraint optimization problem is NP-hard [18], we have developed two heuristic algorithms that are used by the QMEX AS to achieve near optimal total user satisfaction, subject to specified constraints under limited system resources across multiple concurrent heterogeneous sessions. The *Iterative MOS increase algorithm* (Algorithm 1), based on the idea in [13], executes in two phases. During the first phase, the algorithm allocates such amount of resources for each session to achieve minimal acceptable MOS as specified by the user. Based on such given calculated operating points, throughout the second phase, the algorithm allocates the remaining resources until they are all exhausted. At each iteration step, the algorithm searches for the system flow which would contribute most to the global objective function by switching from the current operating point to the one with equal or higher value of MOS. Once all options and/or resources are exhausted, the algorithm terminates. The term $\Delta MOS_{u,i,(j,j+1)}$ represents the difference in the MOS value between two consecutive operating points of the same flow. Similarly, $R_{u,i,(j,j+1)}$ represents resource consumption between the two operating points. The resources shared among multiple user flows are given by a multidimensional vector, where its elements are representing the available bandwidth in different network QoS classes. $R_{u,i,j}$ is thus defined as the vector norm of the bandwidth consumption of the operating point *j* relative to the total amount of bandwidth resource in each of the QoS class. Moreover, $R_{u,i,(j,j+1)}$ is calculated as the difference between $R_{u,i,j+1}$ and $R_{u,i,j}$. A switch to the operating point consuming resources in the QoS class with less capacity will therefore yield greater $\Delta R$ value.

**Algorithm 1** Iterative MOS Increase algorithm.

```
for all user $u \in U$ do
    assign to each flow $i$ operating point satisfying minimum MOS requirement;
end for
$sort$(operating points by their $MOS$ value);
repeat
    for all flow $i$ of user $u$ do
        evaluate utility gain $\Delta_{u,i,(j,j+1)} \leftarrow \frac{\Delta MOS_{u,i,(j,j+1)}}{\Delta R_{u,i,(j,j+1)}}$;
    end for
    create set $S = \Delta_{u,i,(j,j+1)}$
    repeat
        choose flow $i$ with the highest $\Delta_{u,i,(j,j+1)}$
        if user capabilities OR network/provider restrictions violated then
            remove flow $i$ from further evaluation (from set $S$);
        else
            update the operating point of the chosen flow $i$;
            evaluate next utility gain $\Delta_{u,i,(j+1,j+2)}$ and place it the set $S$;
        end if
    until no flow updated AND set $S$ not empty
until (flow from S updated)
```

The idea for the second algorithm, called *Adapted Greedy algorithm* (Algorithm 2), is based on [9]. The main concept is to perform initial distribution of network and provider resources across all system flows and later perform series of switches of resources between the flows to increase the value of the overall utility. The algorithm in [9] suggests a fair initial distribution, where each flow is allocated the same amount of resources. This approach seems reasonable when the flows belong to a single media type such as given by [9]. In our case, however, we consider heterogeneous flows, where audio, video, and download sessions compete for resources; and different media types typically consume

**Algorithm 2** Adapted Greedy algorithm.

```
for all user $u \in U$ do
    calculate average resource consumption;
end for
assign to each user $u$ amount of available network resources according to his average resource consumption;
for all user $u \in U$ do
    find the optimal solution within given resource;
    for all flow $i$ of user $u$ do
        $sort$(operating points by their $MOS$ value);
        create list $L_{u,i}$ containing sorted operating points
    end for
end for
repeat
    for all pairs of flows $i$ and $i'$ do
        calculate $\Delta_{(}i, i+1) \leftarrow \frac{\Delta MOS_{u,i,(j,j+1)}}{\Delta MOS_{u',i',(j',j'-1)}}$;
    end for
    while $\exists s \in S \mid s > 1$ do
        choose max element $s$ of the set $S$;
        if user capabilities OR network/provider restrictions violated then
            remove element $s$ from further evaluation (from set $S$);
        else
            update the operating points of chosen flows $i$ $(j \rightarrow (j+1))$ and $i'$ $(j' \rightarrow (j'-1))$;
        end if
    end while
until gain in global utility achieved $>$ optional threshold
```

different amounts of resources. For example, an audio flow consumes less bandwidth than a 1080p video flow. Hence, our initial resource distribution is based on the average value of resource consumption per flow, based on media type. This approach results in fewer switches and a reduced algorithm execution time.

The ratio $\frac{\Delta MOS_{u,i,(j,j+1)}}{\Delta MOS_{u',i',(j',j'-1)}}$ represents the relative improvement in global utility function if flow *i* is assigned an operating point achieving greater value of MOS and flow $i'$ is assigned with the operating point achieving lower MOS than the currently selected one. Since operating points achieving higher values of utility typically consume more system resources, performing the switch from the operating point $j$ to $(j+1)$ of flow $i$ and from the point $j'$ to the point $(j'-1)$ of the flow $i'$ will result in reallocation of resources from the flow $i'$ to the flow $i$. Both algorithms represent heuristic approaches that typically cannot achieve a global optimum. In addition, to find the optimum, the execution time might be prohibitively long. Therefore, we enable the algorithms to terminate after a preset *Search Depth* parameter. This parameter implicitly scales the size of the switch pool $S$. In the case of *Iterative MOS Increase algorithm*, search depth denotes the number of points of each flow from the list $L_{u,i}$ which will be considered for potential switch from the currently selected one, while in the case of *Adapted Greedy algorithm*, this parameter stands for the number of flow operating points in both upwards and downwards direction from the currently selected point considered for potential upgrade and degradation, respectively. Both algorithms may be upgraded to perform a retest of operating points' feasibility discarded from further evaluation at some point during algorithm execution.

## VI. Evaluation and example

The algorithms are implemented in Java programming language and executed under Sun Java Runtime Environment SE 1.6.0_07-b06 using an Intel Core2 Duo T5250 @ 1.50 GHz with 2 GB RAM and running Microsoft Windows XP Service Pack 3. Five groups of user sessions representing different types of offered services have been considered: 1) audiovisual conference, 2) VoIP call, 3) file sharing, 4) movie download (with subtitles), and 5) simple interactive game. Each service is represented by a different number and type of media flows (audio, video, and data). Quality levels for operating points based on the achieved value of MOS include: HIGH: 4.2–4.5, MEDIUM: 3.9–4.2, and LOW: 3.5–3.9. The maximum number of flow operating points at each quality level is set to two. Total network resources (in uplink and downlink direction) are modeled as a linear function of number of session requests. Each of the five services is hosted by two service providers, SP1 and SP2. The number of users pertaining to each of the two providers is the same, while the resources given to SP1 in the QoS class of the highest quality (the lowest value of delay, jitter and PEP parameters) are set to be 40% less than the resources given to SP2. Consequently, sessions belonging to users served by SP1 achieve a lower value of global MOS. The decision mechanism determining which provider to allocate for which user is outside the scope of this article. Similar to the network resources limitation, values of maximum resources available to each provider are formulated as a linear function in number of sessions.

We uniformly increase the number of session requests for each of the five service groups and recalculate the optimal service configuration for each session. Constraints taken into account during optimization procedure are network and service provider available bandwidth resources, minimal value of session MOS stated by the user, user terminal capabilities in terms of maximum available memory, CPU, downlink and uplink bandwidth connection, and finally, user budget. Numerical values of these constraints are intentionally set in such a way to make allocation of operating point achieving the highest MOS for each system flow unattainable. This testing scenario has been applied for both previously described algorithms.

To obtain reference values, the optimal solution was calculated by using GNU Linear Programming Kit (GLPK; available at http://www.gnu.org/software/glpk/). Due to the size of the optimization problem and the inherent complexity of mixed integer linear programming algorithms, GLPK soon reached a limit for producing a result in a reasonable amount of time for our purposes. For example, for 15 user sessions, the GLPK solver generated the result in approximately 520 seconds (over 8 minutes), which would not be applicable in a real scenario (like during session establishment). Thus, for up to 15 user

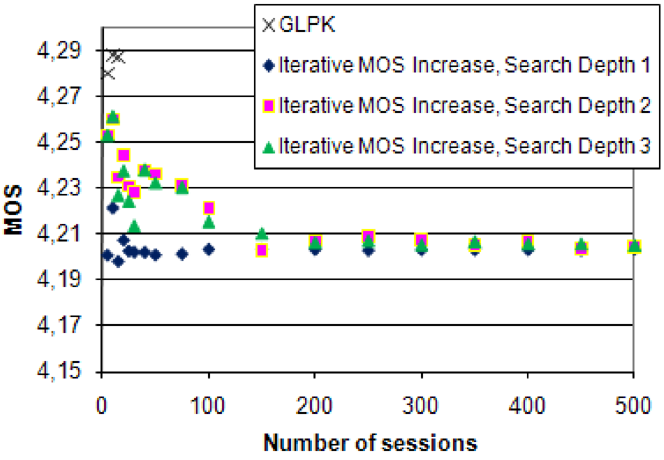

Fig. 2. Global MOS achieved by Iterative MOS Increase algorithm.

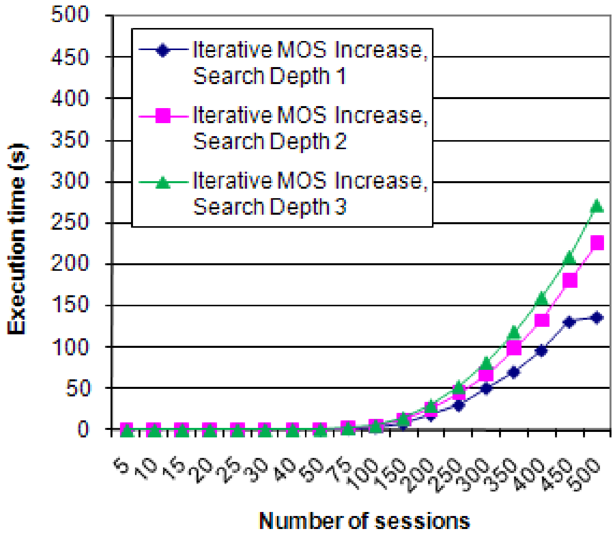

Fig. 3. Execution time of Iterative MOS Increase algorithm.

sessions, the outcomes of the two algorithms are compared with each other as well as with those obtained by using GLPK, and for a larger number of user sessions with each other only.

Figures 2 and 3 show the MOS and execution time, respectively, for the *Iterative MOS Increase algorithm*. It may be noted that the algorithm demonstrates fairly good MOS performance and that it runs much faster than GLPK for multiple user sessions. The achieved value of global MOS is within 98.1% of the optimal solution obtained for a smaller number of sessions by GLPK solver. Due to a possibility of encountering a local optimum, differences in utility values may appear in results for a small number of sessions when the Search Depth parameter is set at 2 and 3. One can further see that for a larger number of user sessions, the *Search Depth* parameter does not contribute to a significant improvement in the global utility value. Increasing the value of this parameter improves the overall utility function only at the third decimal point while, on the other hand, the execution time increases considerably.

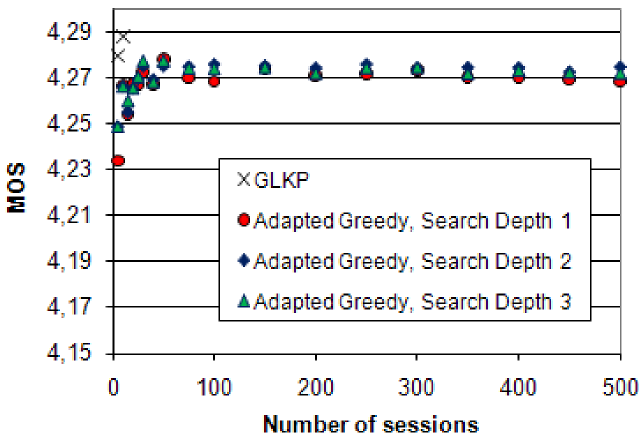

Fig. 4. Global MOS achieved by Adapted Greedy algorithm.

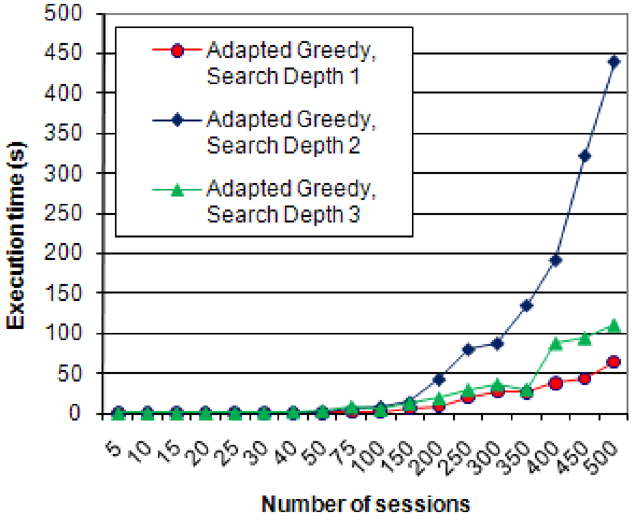

Fig. 5. Execution time of Adapted Greedy algorithm.

Figures 4 and 5, show the MOS and execution time, respectively, for the *Adapted Greedy algorithm*. The achieved value of global MOS is within 99% of the optimal solution obtained for smaller number of sessions by GLPK solver. In terms of execution time, the *Adapted Greedy algorithm* outperforms the GLPK solver with a factor of 100 for 10 concurrent sessions, while for the case of 15 sessions the factor exceeds the value of 3000. As mentioned earlier, the GLPK solver needs about 520 seconds for the latter, while our algorithm (implemented in Java and not optimized as such) requires between 150 ms and 190 ms, even for 100 sessions.

It may also be noted that while the execution time progressively increases with the increase in number of sessions and search depth, as expected, the relationship is not as "regular" as that found in the *Iterative MOS Increase algorithm*. Since the set $S$ expands proportionally to the value of the search depth parameter, it may be expected that for a larger search depth parameter (and a larger "search area"), the algorithm would yield ever better results in longer execution time. This, however, is not the case because the algorithm makes the biggest relative improvement in MOS when selecting the greatest element of the set $S$. With search depth set to 3, it will achieve (again, relatively) larger contributions to the overall MOS at the beginning of its execution, while the smaller contributions thereafter will be disregarded (since a minimal utility gain threshold is imposed). On the other hand, with the search depth set to 2, these utility gains at each step are smaller than in the previous case (e.g., in order to switch from the operating point $j$ to operating point $j+3$, it takes two steps with search depth 2, and one step with the search depth 3). This explains why the execution time for search depth 2 is higher than that with search depth 3. Also, the aberrant behavior of execution time at the search depth 1 in Fig. 3 when the number of sessions exceeds 400, and at search depth 3 in Fig. 5 when the number of sessions exceeds 350, may be ascribed to the particular data used in this example.

The key point is to observe both the achieved MOS and the execution time for a larger number of sessions, and to note that the gain in MOS when using a higher *Search Depth* parameter is actually insignificant in practical terms (between 1.5% and 0.005%). Hence, the higher values of the search depth value only (unnecessarily) increase the execution time in both algorithms without achieving significant MOS gain.

To conclude, the *Adapted Greedy algorithm* may be considered to perform better than the *Iterative MOS Increase algorithm*, both in the terms of the overall MOS value and the execution time. It is also worth noting that the difference in terms of global utility between both algorithms and the optimal solution is very small, even when limiting the search depth to 1. Therefore, both algorithms produce acceptable operational points where users are not penalized in achievable quality. The execution time, however, for the *Adapted Greedy algorithm* is significantly lower for a large number of sessions and limited search depth. Thus, in the context of deployment as part of session setup in the IMS system, the *Adapted Greedy algorithm* would be a better option.

We have also tried versions of the two algorithms performing a retest of the switches to the operating point of some flow which was pronounced infeasible at some point during the execution process, but this did not result in any improvement in the overall utility.

## VII. Conclusions and future work

In this work, we have designed, implemented, and evaluated a new functionality for the QMEX AS in the IMS, which allows to negotiate and optimize QoE parameters in multi-user heterogeneous multi-session multi-media scenarios. Based on a described mathematical model, the QMEX AS tries to optimize overal user satisfaction given various constraints. As the resulting optimization problem is NP-hard, we have developed two heuristic algorithms which can find configurations for the multimedia sessions that maximize a global utility function based on MOS modeling of user perceived quality (under given resource and fairness constraints) within the set of feasible service configurations. The proposed algorithms perform well in scenarios consisting of fewer than 100 concurrent sessions. Here, the execution time was in the order of below 150–190 ms, while computing operational points that yielded near optimal overall achieved MOS. Invoking optimization algorithms in order to provide adequate admission policies represents an issue to be addressed in future work. Moreover, expanding QMEX AS adaptation mechanisms in dynamically changing network, service, or user conditions by performing global resources reallocation is also a topic for further study.

## Acknowledgements

This work was initiated within past COST Action 290 Wi-QoST STSM, and completed within COST Actions IC0703 TMA and IC0906 WiNeMO. Partial support of MZOS RH project 036-0362027-1639 is also gratefully acknowledged.

## References

[1] G. Camarillo , M.-A. Garcia-Martin, “The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds”, Wiley, 2008.

[2] L. Skorin-Kapov, M. Mosmondor, O. Dobrijevic, M. Matijasevic. “Application-Level QoS Negotiation and Signaling for Advanced Multimedia Services in the IMS”, *IEEE Communications Magazine*, Vol. 45, No.7, pp. 108-116, July 2007.

[3] L. Skorin-Kapov, M. Matijasevic, “A QoS Negotiation and Adaptation Framework for Multimedia Services in NGN”, *Proc. of 10th Intl. Conf. on Telecommunications*, pp. 249-256, Zagreb, Croatia, June 2009.

[4] L.-U. Choi, W. Kellerer, E. Steinbach, “Cross layer optimization for wireless multi-user video streaming”, *Proc. Intl. Conf. on Image Processing*, vol. 3, pp. 2047-2050, Singapore, October 2004.

[5] X. Liu, E. K. P. Chong, N. B. Shroff, “Transmission scheduling for efficient wireless utilization”, *Proc. 20th IEEE INFOCOM’01*, vol. 2, pp. 776-785, Anchorage, Alaska, USA, April 2001.

[6] ITU-R Recommendation BT.500-11, “Methodology for the subjective assessment of the quality of television pictures”, June 2002.

[7] Video Quality Experts Group, “Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment”, [On-line: www.vqeg.org], October 2003.

[8] ITU-T Recommendation P.910, “Subjective video quality assessment methods for multimedia applications”, April 2008.

[9] S. Khan, S. Duhovnikov, E. Steinbach, W. Kellerer, “MOS-Based Multiuser Multiapplication Cross-Layer Optimization for Mobile Multimedia Communication”, *Advances in Multimedia*, Hindawi Publishing Corp., Vol. 2007, Article ID. 94918, 11 pp., June 2007.

[10] D. Jurca, P. Frossard, “Media-Specific Rate Allocation in Multipath Networks”, *IEEE Trans. on Multimedia*, Vol. 9, No. 6, pp. 1227-1240, Oct. 2007.

[11] ITU-T Recommendation G.107, “The E-model: a computational model for use in transmission planning”, May 2000.

[12] D. Jurca, W. Kellerer, E. Steinbach, S. Khan, S. Thakolsri, P. Frossard, “Joint Network and Rate Allocation for Video Streaming over Multiple Wireless Networks”, *Proc. 9th IEEE Intl. Symp. on Multimedia*, pp. 229-236, Dec. 2007.

[13] R. Mahalingam, T. Wei, E. Steinbach, “RD-Optimized Rate Shaping For Multiple Scalable Video Streams”, *Proc. IEEE International Conference on Multimedia and Expo*, pp. 1794-1797, July 2007.

[14] R. Rajendran, S. Ganguly, R. Izmailov, D. Rubenstein, “Performance Optimization of VoIP using an overlay Network”, *IEEE 25th INFOCOM 2006*, April 2006.

[15] M. Ries, M. Crespi, C. Nemethova, O. Rupp, M., “Content Based Video Quality Estimation for H.264/AVC Video Streaming”, *Proc. of IEEE WCNC*, pp. 2668 - 2673, March 2007.

[16] H. R. Sheikh, M. F. Sabir, A. C. Bovik,“A statistical evaluation of recent full reference image quality assessment algorithms”, *IEEE Trans. on Image Processing*, vol. 15, no. 11, pp. 3440-3451, Nov. 2006.

[17] K. Brunnstrom, D. Hands, D., F. Speranza, A. Webster, “VQeg validation and ITU standardization of objective perceptual video quality metrics [Standards in a Nutshell]”, *IEEE Signal Processing Magazine*, Vol. 26, No.3, pp. 96-101, May 2009.

[18] S. Dasgupta, C. Papadimitriou, U. Vazirani, “Algorithms”, McGraw-Hill, 2006.