Utility Based Model for Optimized Resource Allocation for Adaptive Multimedia Services

Krunoslav Ivesic, Maja Matijasevic, Lea Skorin-Kapov
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, Zagreb, Croatia
{krunoslav.ivesic, maja.matijasevic, lea.skorin-kapov}@fer.hr

***Abstract*—The problem of dynamic Quality of Service (QoS) management for multimedia services has been addressed from two complementary, yet interrelated directions; one mainly dealing with designing adaptive or "elastic" services, and the other, dealing with "smarter" resource allocation. We adopt this approach in this paper, by performing a graceful quality degradation and the corresponding resource (re)allocation for multiple multimedia sessions in case of decreasing network resources. We introduce a mathematical model which optimizes overall system utility by taking user satisfaction and operator profit into account.**

I. Introduction

From the point of view of the network provider, a multimedia service combines two or more *media components*, such as audio, video, text, data, 2D/3D graphics, etc. [1]. Within a single session, the media components may be combined in different ways, and need not be active at the same time – their number, as well as properties and requirements in terms of network resources typically vary. The problem of dynamic Quality of Service (QoS) management for such services has been addressed from two complementary, yet interrelated directions; one mainly dealing with designing adaptive or ”elastic” services, and the other, dealing with ”smarter” resource allocation.

In this work we present our early work in modeling resource allocation, considering multiple sessions within a domain under control of a single network provider. Our particular interest is the situation when available network resources decrease and a degradation of active sessions is due to occur. We present the method for "graceful degradation" of established sessions with service priority, user category and operator's profit taken into account. As the basis for service degradation we utilize the concept of Media Degradation Path (MDP) introduced in our previous work [2][3]. The MDP specifies possible service configurations along with their resource requirements and utility values. Since the MDP contains the "knowledge" about the service, it enables the dynamic adaptation and the possibility of negotiation. The degradation to one of the previously agreed configurations from the MDP is likely be more acceptable to the end user than the unpredictable change with uniform probability of degradation for each service flow.

With the MDPs specified for sessions it is possible to perform optimization, i.e. to optimally distribute available resources for all active sessions in the domain. In case of the decrease in available network resources some services have to be degraded, based on their priorities. Less demanding configurations from these services' MDPs are activated and overall system utility is kept as high as possible.

This paper is organized as follows. In section II we briefly discuss the related work. Section III sums up state of the art in provision and adaptation of rich multimedia services. Section IV presents our optimization model. The experimental scenario is presented in Section V. Section VI concludes the paper and outlines the future work.

II. Related Work

The problem of management of multiple multimedia sessions with various possible QoS configurations was described in [4] and mathematically formulated as a multi-choice multi-dimensional 0-1 knapsack problem (MMKP), which is known to be NP-complete [5]. For that reason the exhaustive search algorithms for the solution to the problem would be inefficient (as showed later in section IV) and heuristic algorithms need to be developed. The standard heuristics for approximate solutions, like genetic algorithms, simulated annealing and tabu search turned out to be costlier [6] than dedicated heuristics developed specially for MMKP, like [5] [6] [7] [8].

We assume that the degradation should be performed with particular application and its state taken into consideration. By using the MDP, the service endpoints can agree on possible configurations and the service can be switched to a less demanding configuration in case of the decrease in the available network resources.

III. Model Background

In order to support multimedia service delivery over a multi-access converged all-IP core network, the Third Generation Partnership Project (3GPP) has finalized the Release 8 specifications of the Evolved Packet System (EPS) [9]. The EPS specifies class-based QoS provisioning, allowing operators to differentiate the treatment received by different subscribers and services. Functional network entities and interfaces responsible for providing service-aware QoS control have been specified as part of the overall 3GPP Policy and Charging Control (PCC) architecture [10]. The key concept introduced by EPS is the QoS Class Identifier (QCI), which provides a standardized reference for packet forwarding treatment in a given session data flow. The 3GPP specifications include nine QCIs, where each QCI specifies:

- priority – a number from 1 to 9; with 1 being the highest priority,
- bearer type (GBR or non-GBR),
- packet delay budget (PDB),
- packet error loss rate (PELR).

In addition to QCI, the Allocation and Retention Priority (ARP) [10] specifies control plane-treatment for bearers, i.e., it may be used to decide whether a bearer establishment or modification request should be accepted or rejected due to resource limitations. The ARP specifies:

- priority – a number from 1 to 15; 1 being the highest priority,
- pre-emption capability (yes or no),
- pre-emption vulnerability (yes or no).

The pre-emption capability defines whether the bearer can acquire the resources already given to the bearer with lower priority, and the pre-emption vulnerability defines whether the bearer can lose the resources in favor of the bearer with higher priority. The decision on how to assign these parameters to the bearers is left to the operator. It can depend on user subscription category and service category.

A. Media Degradation Path

The idea of the MDP utilizes the modified concept of a mapping between adaptation, resource and utility spaces, defined for each media component and introduced in [11]. The adaptation space is defined as the space of all possible adaptations of media, e.g. frame-rate and resolution for video. The resource space is defined by the required resources, e.g. bandwidth and required memory. The utility space is the space of quality measures, e.g. temporal smoothness and audio-visual rhythm. Our modification in [2] restricts the adaptation space to an operating space that consists of only those adaptations, the parameters of which are agreed end-to-end. This modified mapping has been named Operating-Resource-Utility (O-R-U) mapping. The points in O are mapped to points in R and U. There can be several points in O that can be mapped to the particular subset of R that corresponds to the resource requirements that the UE can meet. These points can have different mappings in U and the goal is to select those points from O that maximize the utility, which is defined as a function of multiple dimensions of U. Upon selection of the optimal points that represent the best parameters sets for the media contained in the particular session, the optimal configuration for that session is created. In addition, several next-to-optimal points are selected for other acceptable parameters sets, thus creating several alternative configurations. The created configurations are ordered by the decreasing utility value and collected into a new MDP.

The MDP can be utilized for parameters negotiation and management of multimedia services. When user requests a service, the service endpoints, e.g. the user equipment (UE) and the application server (AS), start negotiating required service resources. In case of a service composed of several media flows it is expected that several bearers will be required, either occasionally or during the whole session duration, in order to support the provision of the individual service components that can be dynamically added or removed during the service lifetime. During negotiation several service configurations are agreed on. It is likely that these configurations differ in number of included media components. While the service is provisioned it is possible to switch to another previously agreed configuration due to change in service requirements, user action or change in network resources.

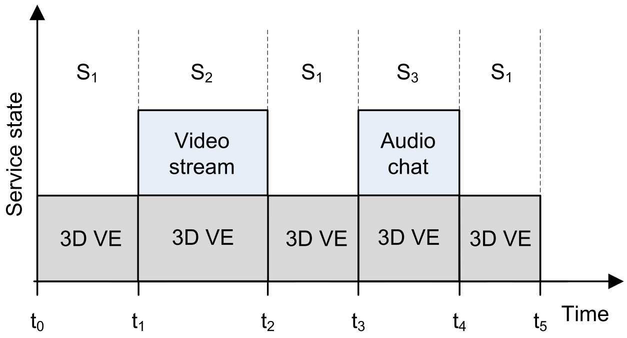


Fig. 1. An example scenario of a service with three states

In this work we extend the fundamental idea of the MDP by introducing service dynamics in form of service *states*. By the service state we assume a set of service components that are simultaneously active during a particular time interval during service provisioning. Fig. 1 depicts a possible running scenario of an example service with with three states. The service consists of a continuously presented three-dimensional graphics virtual environment (VE) with possibility of adding video stream or audio chat. The state S_1 denotes the time intervals when only VE is active. The states S_2 and S_3 are entered by adding video stream and audio chat respectively. It is important to note that the “transition” between different states is not scheduled in advance and Fig. 1 shows only one possible scenario where the state S_1 is active initially from the time t_0 till the time t_1. From t_1 to t_2 the active state is S_2 because the video stream is active. By stopping the video at t_2 the state S_1 is entered again and it lasts till t_3, when audio chat is activated and the state S_3 is entered. The audio chat is active till t_4 and then the state S_1 is entered again and is active till t_5, at which the service ends.

Different states of the service that demand different resources should have separate groups of configurations. Switching to another configuration caused by the change in service requirements should be treated differently from the degradation (or upgrade) to another service configuration due to the change in network resources. So, it is desirable to group configurations based on the state of the service they belong to. Fig. 2 depicts this concept. For different service states different configurations are defined. For example, for the service the sample scenario of which is shown on Fig. 1, the states S_1, S_2 and S_3 would require three different groups of configurations. For the state S_1 the configurations would be concerning only VE presentation quality, e.g. level of details, whilst for the

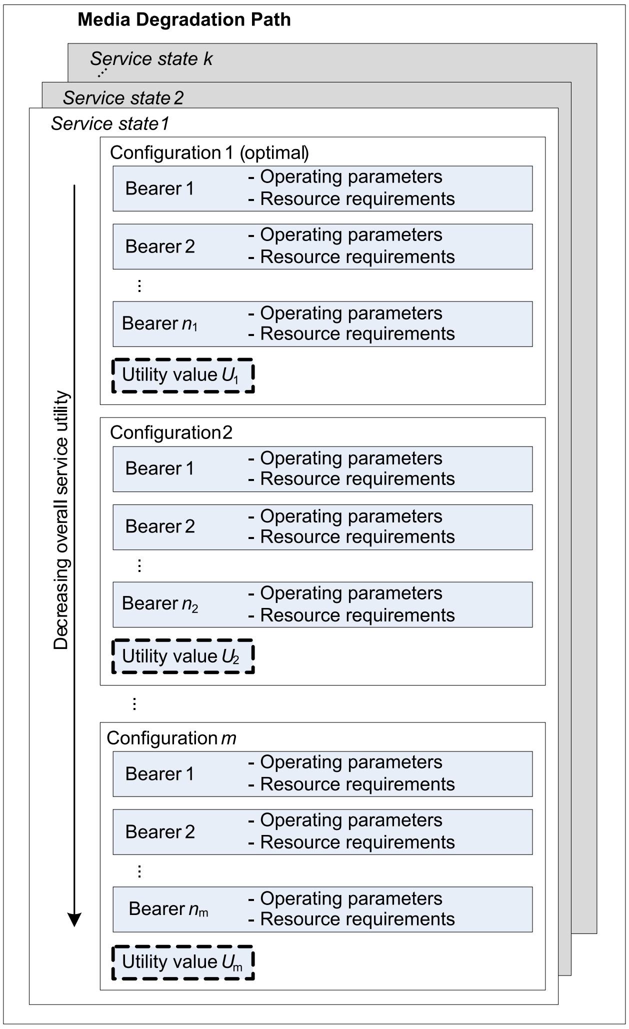


Fig. 2. Media Degradation Path

states S_2 and S_3, the configurations would also include video or audio quality parameters, e.g., frame-rate, bit-rate, etc.

B. Mapping the Problem to MMKP

With the assumption that only a single service state is active at any given time, only the set of configurations belonging to that state needs to be considered. The objective of the optimization is to select a configuration for each service so that the overall utility and operator's profit are maximized.

Let n be the number of currently active sessions in the domain and p_u the number of configurations of the session u (for currently active state of session u). The configuration i of session u has z_{ui} media flows, of which flows $1, ..., h_{ui}$ are in the downlink direction, and flows $h_{ui}+1, ..., z_{ui}$ are in the uplink direction. The bandwidth requirements of particular configuration are denoted by the vector $\vec{b}_{ui} = (\vec{b}_{ui1}, ..., \vec{b}_{uiz_{ui}})$. For session u and configuration i, each media flow j has requirements specified by the vector $\vec{b}_{uij} = (\vec{b}_{uij1}, ..., \vec{b}_{uij9})$ which describes bandwidth requirements on QCIs from 1 to 9 for flow j. We assume that utility values are specified for each configuration and we denote them as $U_{ui}(\vec{b}_{ui})$. In order to be able to compare the utilities of configurations belonging to different sessions, it is necessary to normalize the utility values. The configurations for the particular state are sorted by decreasing utility value, so that the first configuration has the highest utility value. Normalized utility values are thus calculated by dividing the utility values of a state by the utility value of the first configuration:

$$U\_n_{ui}(\vec{b}_{ui}) = \frac{U_{ui}(\vec{b}_{ui})}{U_{u1}(\vec{b}_{u1})} \tag{1}$$

In order to differentiate the users we specify weight factors for user category as $w_u^{category}$ and for service category as $w_u^{service}$. The session priority is then defined as a product of these two weight factors:

$$w_u = w_u^{category} \cdot w_u^{service} \tag{2}$$

Operator's revenue and cost are calculated per configuration and denoted as $R_{ui}(\vec{b}_{ui})$ and $C_{ui}(\vec{b}_{ui})$ respectively. The profit is then calculated as the difference between the two values. The profit value is also normalized within each state to ensure fair consideration of different sessions. Maximum allowed bandwidth for downlink and uplink directions are denoted as B_{k_DL} and B_{k_UL} respectively, where k is the QCI number. If we want to maximize the overall utility value and the operator's profit, their respective objective functions, F_{ut} and F_{op}, can be formulated as:

$$F_{ut} = \sum_{u=1}^{n} \sum_{i=1}^{p_u} \left\{ w_u x_{ui} U\_n_{ui}(\vec{b}_{ui}) \right\} \tag{3}$$

$$F_{op} = \sum_{u=1}^{n} \sum_{i=1}^{p_u} w_u x_{ui} \frac{R_{ui}(\vec{b}_{ui}) - C(\vec{b}_{ui})}{\max_i \left[R_{ui}(\vec{b}_{ui}) - C(\vec{b}_{ui}) \right]} \tag{4}$$

Binary variables x_{ui} are added to indicate selected configurations. These two objective functions need to be maximized and thus form the multi-objective optimization problem which can be converted to a single-objective problem by adding the two functions multiplied by their respective weight factors, $w_{utility}$ and w_{profit}. The problem can now be formulated in the following way:

$$max(w_{utility} F_{ut} + \beta w_{profit} F_{op}) \tag{5}$$

such that:

$$\sum_{u=1}^{n} \sum_{i=1}^{p_u} \sum_{j=1}^{h_{ui}} x_{ui} b_{uijk} \le B_{k\_DL}, k = 1, ..., 9 \tag{6}$$

$$\sum_{u=1}^{n} \sum_{i=1}^{p_u} \sum_{j=h_{ui}+1}^{z_{ui}} x_{ui} b_{uijk} \le B_{k\_UL}, k = 1, ..., 9 \tag{7}$$

$$\sum_{i=1}^{p_u} x_{ui} = 1, u = 1, ..., n \tag{8}$$

$$x_{ui} \in \{0, 1\}, u = 1, ..., n, i = 1, ..., p_u \tag{9}$$

The constant β in expression (5) is added for measurement units conversion because the utility part and the profit part can be expressed in different units.

In our model overall utility values (expression (3)) and operator profit (expression (4)) are calculated based on resource requirements of particular configuration, which introduces correlation to the data set. It is known that correlation adds additional complexity to the MMKP problem [6]. However, it is expected that in a real network this will also be the case.

IV. Optimization of Multiple Sessions in Mathematica

To test various scenarios in the optimization of multiple sessions we developed a model in Wolfram *Mathematica* 7.0. For a given number of parallel sessions our algorithms define several configurations for each session. The number of configurations per session is randomly chosen from a preconfigured interval. For the first configuration of each session the number of used QCIs is defined, as well as QoS requirements for these QCIs. For simplicity we assumed that every service has pre-emption capability and pre-emption vulnerability parameters set to "yes". Other configurations are calculated by randomly decreasing QoS requirements or even discarding some QCIs in subsequent configurations. For example, for a service with three configurations the first one is defined with random QoS requirements, the second one could use 80% of the QoS requirements of the first one, and the third one could remove one QCI and further decrease the second configuration by retaining 75% of QoS requirements. After definition of QoS requirements for all sessions utility and revenue values are calculated for each configuration. Utility values are calculated based on QoS requirements of each configuration and then normalized, as explained in the previous section. Operator cost is also defined based on QoS requirements and the number of used QCIs. The service price that the end user pays is calculated based on the number of QCIs used and requested QoS parameters. Revenue is calculated as the difference between the price and the cost. Having all configurations defined, the requirements of the best configurations from all sessions are summarized which defines the maximal requirements on QCIs for the currently generated sessions. The QoS limits used for the simulation of the degradation of network resources are generated as a list consisting of maximal requirements on QCIs and limits which are calculated by multiplying these maximal QoS requirements by factors 0.9, 0.8, ..., 0.4, thus enabling limiting QoS to 90%, 80%, ..., 40% of maximal required values.

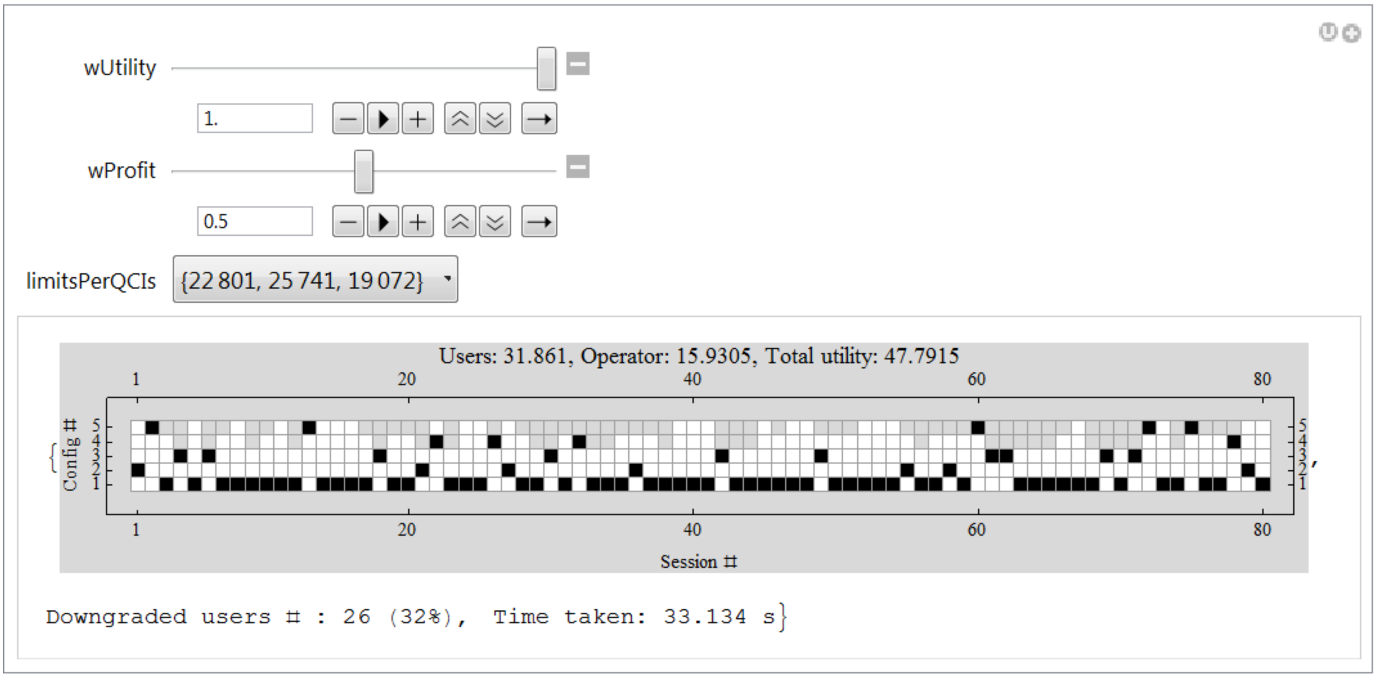
Having all required parameters calculated, the graphical representation of sessions is displayed in a GUI, as depicted by Fig. 3. It consists of control elements and visualization of the optimization result. The control elements are used to manipulate the parameters that affect the optimization: weight factors $w_{utility}$ and w_{profit} and QCI limits (in this example, we considered three QCIs). The weight factors can be assigned the values from 0 to 1 in steps of 0.1 and the the QCI limits can be set to any value from a predefined list, as explained above. The visualization result presents all sessions in the system in form of a matrix (considering only currently active service states). The horizontal axis represents sessions and the vertical axis represents their configurations. Each session in represented by one column. The number of squares in the column corresponds to the number of configurations in the active state of the session in question. Black square represents the currently active configuration, whilst white squares represent other configurations, which are currently inactive. There is exactly one black square per column because only one configuration per session is active at any given time. The configuration are sorted by utility; thus the configuration 1 is "better" (i.e., it has higher utility) than the configuration 2, the configuration 2 is better than the configuration 3, and so on. Thus, the position of the black square in the column indicates how favorable (with respect to the utility of the session represented by the column) the selected configuration is: the lower the black square is positioned within the column, the better the selected configuration is. The label above the matrix consists of the values of the functions F_{ut} (Users), F_{op} (Operator) and the value of the objective function (Total utility). Below the matrix the statistical data regarding the last instance of the optimization process is written: the number of downgraded users and the time taken to calculate the solution.

By using the GUI, the optimization process is initialized by moving the sliders or changing the QCI limits – the objective function (expression (5)) is maximized again with new weight factors and QCI limits, possibly resulting in redistribution of resources causing switches of active configurations. This is demonstrated as a redistribution of the black squares in the matrix.

V. Evaluation of the Model

The built-in maximization algorithm used in *Mathematica* guarantees to find the global maximum for the linear problem, which is the case in this work. This approach produces the optimal solution, but it is too time consuming, e.g., for the problem shown on Fig. 3, the optimization process took 33.134 seconds. The test was performed on a PC with Intel Core Quad Q9400 2.66 GHz, 4 GB RAM running on 64-bit Microsoft Windows 7. For the real-time services, (calls, conferences, etc.) it is necessary to conduct the optimization process in terms of milliseconds and signalize the required changes to the sessions endpoints rather quickly, because the adaptation of the ongoing sessions should not take longer than 1 second, as has been proposed in [12]. As mentioned earlier, the heuristic algorithm has to be developed. This model can therefore be used as a comparison to a heuristic algorithm in terms of quality of calculated solutions.

On Fig. 4 an example of gradual degradation of 60 sessions is shown. The first image (a) depicts the initial situation when the resources are sufficient to provide the maximal required QoS for all active sessions and no session is degraded, which is demonstrated by the fact that the first (optimal) configuration of each session is active, i.e. all black squares are in the lowest row of the matrix. On the second image (b) the resources have decreased to 90% and a few sessions have been degraded, but the majority of sessions have not been affected. This is demonstrated as a few black squares raised to the higher rows.



wUtility 1.

wProfit 0.5

limitsPerQCIs {22 801, 25 741, 19 072}

Users: 31.861, Operator: 15.9305, Total utility: 47.7915

Downgraded users ♯ : 26 (32%), Time taken: 33.134 s}

Fig. 3. GUI in *Mathematica*

The third image (c) depicts degradation to 70% and it is clear that many sessions have been affected because many black squares have "climbed up". The last image (d) shows the ultimate degradation scenario, i.e. degradation to only 40% percent of maximal required QoS. It is clear that most of the sessions have been affected and that the majority of black squares is in the upper region of the matrix. It may also be noted that the overall system utility is rather low, as it has been degraded from 43.744 in case (a) to 21.554.

In terms of performance, it is hard to provide the estimated time required for the optimization process to run because the problem complexity varies for each generated data set. We believe this is due to varying fitness of different problem instances to exhaustive search algorithms. Sometimes it is possible to prune a lot of partial solutions and sometimes this is not the case. However, all problem instances show the following interesting common feature. For a small decrease in resource availability (90% of max. resources), as well as for extreme decrease (40% of maximum resources), the optimization problem is solved in rather short time. On the contrary, for "moderate" decrease (70%) it takes noticeably longer to calculate the solution. The small decrease discards only several possible problem solutions – the optimal case and a few suboptimal cases and leaves a lot of space for solutions that degrade only a small portion of running sessions. On the other hand, the significant decrease introduces a lot of pruning and leaves very little place for search for the solution. Therefore, after removal of all infeasible cases the very few remaing cases take little time for consideration and determination of the optimal solution. We can also confirm that the time taken to calculate the solution for a smaller data set can be longer than the time taken for a bigger one, as reported in [7].

The lower limit for percentage of remaining resources that we used, i.e. 40%, was determined experimentally. It shows that at 40% of maximal required resources for most data sets it is still possible to find suitable configurations for running services. This result, however, can not be generalized as it depends on the input data set.

The ratio of modified sessions per instance of optimization process can be high. All sessions for which the resources are modified require signalization for renegotiation of parameters. In a real network, this involves a certain signaling overhead, and is thus for further study.

VI. Conclusions and Future Work

Multimedia services involving multiple media components will typically occupy several bearers and it will be possible to provide them by using several agreed configurations. Due to the decrease in available network resources, switching to a less demanding configuration might occur. The proposed model demonstrated the complexity of the mathematical background of the problem and the effects of the change in parameters of the objective function that is maximized. We identified possible intensities of the degradation and issues related to the degradation process. The future work will be based on development of a simulator tool for the optimization of multiple sessions by using a heuristic algorithm.

Acknowledgment

The authors acknowledge the support of the Ministry of Science, Education and Sports of the Republic of Croatia

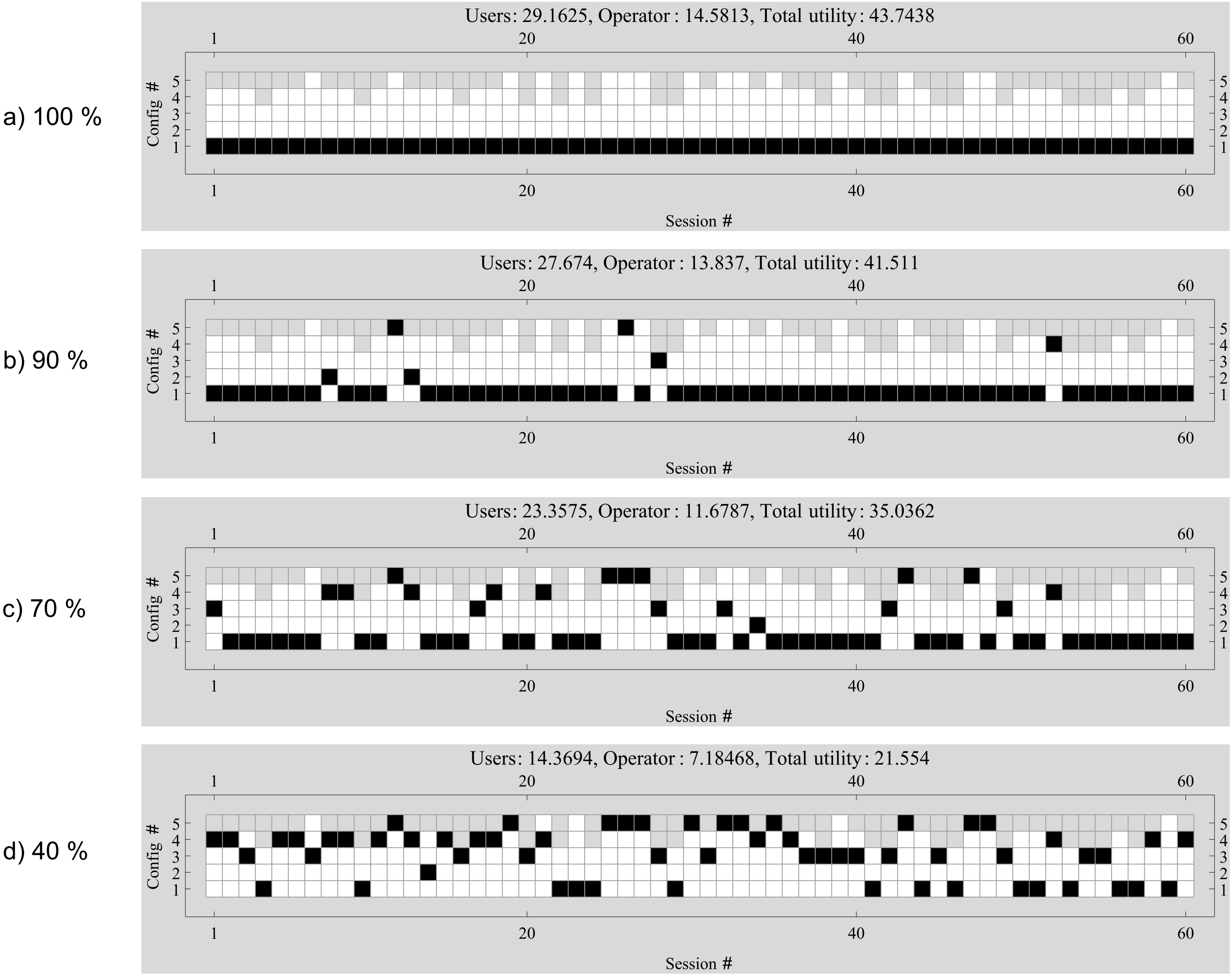


Fig. 4. An example of the gradual degradation

research projects no. 036-0362027-1639 and 071-0362027-2329.

References

[1] —, *Framework Recommendation for multimedia service. ITU-T Recommendation F.700*. International Telecommunication Union, Telecommunication standardization sector, Nov. 2000.

[2] L. Skorin-Kapov and M. Matijasevic, “A data specification model for multimedia QoS negotiation,” in *MobiMedia ’07: Proceedings of the 3rd International Conference on Mobile Multimedia Communications*, (Nafpaktos, Greece), pp. 1–7, ICST, 2007.

[3] L. Skorin-Kapov and M. Matijasevic, “Modeling of a QoS matching and optimization function for multimedia services in the NGN,” in *Proc. of the 12th IFIP/IEEE Intl. Conf. on Management of Multimedia and Mobile Networks and Services, MMNS 2009*, (Venice, Italy), pp. 55–68, Oct. 2009.

[4] S. Khan, K. F. Li, and E. G. Manning, “The utility model for adaptive multimedia systems,” in *In International Conference on Multimedia Modeling*, (Singapore), pp. 111–126, 1997.

[5] M. Moser, D. P. Jokanovic, and N. Shiratori, “An algorithm for the multidimensional multiple-choice knapsack problem,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 80, no. 3, pp. 582–589, 1997.

[6] M. M. Akbar, M. S. Rahman, M. Kaykobad, E. Manning, and G. Shoja, “Solving the multidimensional multiple-choice knapsack problem by constructing convex hulls,” *Computers & Operations Research*, vol. 33, no. 5, pp. 1259 – 1273, 2006.

[7] S. Khan, K. F. Li, E. G. Manning, and M. M. Akbar, “Solving the knapsack problem for adaptive multimedia systems,” *Stud. Inform. Univ.*, vol. 2, no. 1, pp. 157–178, 2002.

[8] H. Shojaei, A. Ghamarian, T. Basten, M. Geilen, S. Stuijk, and R. Hoes, “A parameterized compositional multi-dimensional multiple-choice knapsack heuristic for CMP run-time management,” in *DAC ’09: Proceedings of the 46th Annual Design Automation Conference*, (New York, NY, USA), pp. 917–922, ACM, 2009.

[9] “Service requirements for the Evolved Packet System,” 3GPP TS 22.278, Release 9, Sept. 2009.

[10] “Policy and charging control architecture,” 3GPP TS 23.203, Release 9, Sept. 2009.

[11] S.-F. Chang, “Optimal video adaptation and skimming using a utility-based framework,” in *Proc. Tyrrhenian Intl Workshop on Digital Communications, Capri Island*, 2002.

[12] T. Guenkova-Luy, A. J. Kassler, and D. Mandato, “End-to-end quality-of-service coordination for mobile multimedia applications,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 5, pp. 889–903, 2004.