

Modeling of a QoS Matching and Optimization Function for Multimedia Services in the NGN

Lea Skorin-Kapov¹ and Maja Matijasevic²

¹ R&D Center, Ericsson Nikola Tesla, Krapinska 45, HR-10000 Zagreb, Croatia

² FER, University of Zagreb, Unska 3, HR-10000 Zagreb, Croatia
lea.skorin-kapov@ericsson.com, maja.matijasevic@fer.hr

Abstract. A key challenge for Next Generation Networks (NGN) is providing support for the negotiation and dynamic adaptation of Quality of Service (QoS) parameters. In this paper we propose a novel *QoS Matching and Optimization Function* (Q-MOF) to be included along the signaling path with the purpose of providing advanced service-level QoS matching and optimization logic for multimedia services being delivered to heterogeneous end users and access networks. The Q-MOF is introduced in the NGN architecture as a generic functionality, allowing for it to be reused by services requiring advanced QoS support. We illustrate Q-MOF applicability in a case study involving an audio/video streaming session between two users.

Keywords: QoS matching, QoS optimization, multimedia services, next generation network.

1 Introduction

With the Next Generation Network (NGN) aimed at offering end users personalized multimedia services across heterogeneous networks, a key issue is providing support for end-to-end (E2E) Quality of Service (QoS) negotiation and adaptation (QNA). During QNA, the requirements and capabilities imposed by various actors involved in service delivery (e.g., user, service provider, network provider, third party service/content provider) need to be coordinated [15]. At the service level, NGN standards specify requirements for entities at the rim of the network (e.g., terminals) to negotiate and select common codecs for each E2E session [1]. 3GPP has specified the IP Multimedia Subsystem (IMS) procedures for negotiating multimedia session characteristics between endpoints [3].

With the transition to media-rich services, we argue that providing personalized and optimal service quality implies more advanced negotiation mechanisms (beyond those specified) matching user preferences/capabilities against service requirements, network resource capabilities, and operator policy. Furthermore, in addition to codec negotiation, additional operating parameters (e.g., resolution, frame rate, etc.) may need to be coordinated.

Related work on QoS adaptation in IMS networks includes solutions for dynamically adapting multimedia content to fit network/system resource availability [12] and maximizing perceived QoS [5]. Research on multimedia session

control and content adaptation is further being conducted in the DAIDALOS [18] and ENTHRONE [19] projects. While additional related work can be found on profile matching [9], decision-making for the optimization of service parameters [10][17], and the description of various service and transport configurations for service negotiation [7][13][8], limited solutions unite these aspects to support delivery of personalized multimedia services in the NGN. Our goal is to propose such a solution, with our focus being on service-level negotiation mechanisms.

We have previously proposed the high-level concept of a *QoS Matching and Optimization Function* (Q-MOF) to be included along the E2E signaling path in the service control layer of the NGN architecture [14]. The contribution of this paper is a detailed generic model of the Q-MOF supporting optimized service delivery and controlled service adaptation in light of changing resource availability, user preferences, or service requirements. Furthermore, what is missing in most current approaches is the specification and signalling of an optimal degradation path for multimedia services composed of multiple flows. Our solution is for the Q-MOF to calculate a *media degradation path* (MDP) that will be signaled to communication end points and to the network layer to aid in efficient service adaptation in light of changing network resource availability.

The paper is organized as follows. In Section 2 we describe the role of the Q-MOF in the standardized NGN architecture. In Section 3 we present a model of the Q-MOF, and specify the *Matching Process* and *Optimization Process*. Section 4 describes a case study illustrating Q-MOF applicability for an audio/video streaming session. A discussion and conclusions are given in Section 5.

2 Enhanced Multimedia QoS Control in the NGN

The ITU-T NGN release 1 architecture [1] (Figure 1) is based on the concept of independence between the transport stratum and service stratum. In the service stratum, service control functions are based on an IMS service environment and support the provisioning of real-time Session Initiation Protocol (SIP) based multimedia services. The application support functions and service support functions can impact sessions on behalf of services. During session negotiation, QoS requirements are extracted by service control functions (e.g., Proxy Call Session Control Function in IMS) and used to issue resource reservation and authorization requests to the resource admission control functional architecture (RACF).

In order to provide enhanced QoS control mechanisms, we propose for the Q-MOF to be introduced in the service stratum as a new application server (AS). The Q-MOF is designed to be included along the E2E signaling path, providing support for advanced QoS parameter matching and optimization logic within the QoS negotiation process. Actual service adaptation (e.g., transcoding) may be performed at communication endpoints or in the network.

While today's Internet model is based on a user-centric view of the network, the IMS model uses a more operator-centric approach. In an actual network implementation, we propose for the Q-MOF to be included in a Service Provider (SP) domain as a generic reusable functionality. In addition to providing a better

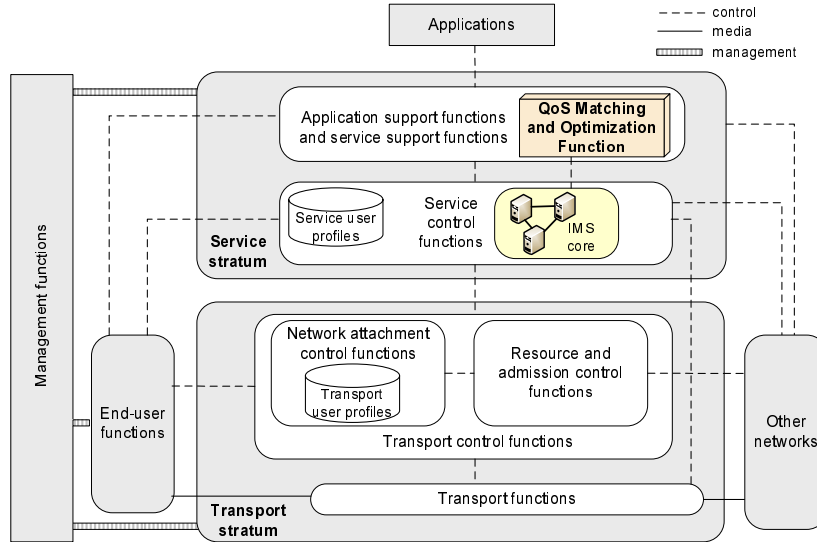


Fig. 1. Q-MOF in the NGN architecture

service to users, introducing enhanced QoS support in the network as a reusable service capability would benefit both the SP and third-party service/application providers. The SP would have additional means to control, differentiate, and appropriately charge the QoS a particular user receives for a given multimedia service. Third-party service/application providers would have to specify a service profile stating service requirements and options and would further be relieved from implementing complex QoS decision making functionality for each new introduced service, hence leading to simplified provisioning and possibly quicker time-to-market for new services requiring such mechanisms.

3 Modeling the Q-MOF

A multimedia service is composed of two or more *media components* (e.g., audio flows, video flows, graphics data, etc.). We assume that a multimedia service exists in one or more versions to meet heterogeneous user and network capabilities. We specify *service versions* as differing in the included media components (e.g., audio/video vs. only audio). Each media component may be configured by choosing from offered alternative *operating parameters* (e.g., different codecs, frame rates, resolutions, etc.). We refer to the overall *service configuration* as the set of chosen operating parameters for all included media components.

The Q-MOF functional architecture is shown in Figure 2. Upon a service request, input data is signaled by involved actors. Various sets of parameters/profiles that may be signaled include: end-user related parameters (e.g., capabilities, preferences, subscription data, context); service requirements, in terms of necessary

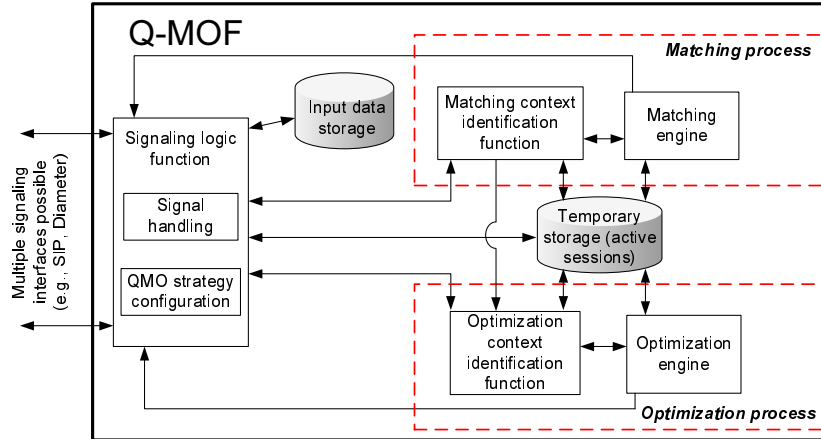


Fig. 2. Q-MOF functional architecture

resources (in the case of an adaptive service then adaptation capabilities need to be specified); network capabilities (e.g., specification of available QoS classes, available bandwidth); operator policies and pricing/charging data. An example of actors and input data is given in the case study. Input data is collected by the Q-MOF (either per-session or collected once and stored to be used for multiple sessions), and a *Matching Process* is invoked to determine a set of feasible service versions and operating parameters termed the *Feasible Service Profile (FSP)*. The FSP is then offered to relevant entities in order to achieve an E2E agreement: an end user may accept, refuse, or modify the offered parameters, and network entities authorize resources. Negotiated and authorized parameters are passed to the *Optimization Process*, which calculates the optimal service configuration and resource allocation for all media flows. The optimal service configuration, and a number of alternative (suboptimal but feasible) configurations are ordered by descending utility and signaled to involved entities as an *Agreed Service Profile (ASP)*. The ordered list of alternative configurations within the ASP is referred to as a *media degradation path (MDP)* to be used for efficient service adaptation.

In order to support the negotiation process, we assume a “one-stop responsibility” concept (adopted by the ITU-T in [2]) with a user’s primary service provider as being responsible for coordinating the QoS negotiation process (via the Q-MOF), while further relying on the services of sub-providers in order to secure an E2E QoS. Therefore, we assume that necessary agreements exist along the multi-provider chain enabling the negotiation of session parameters.

We assume that the Q-MOF can implement multiple signaling interfaces (e.g., SIP, Diameter, etc.) to other services or entities (e.g., policy server, charging server, user/service profile repository). The Q-MOF may be applied for various business models, i.e., it can take into account parameters specified by different actors involved in the service negotiation process. In order for the proposed functionality to be applicable, it is assumed that the entities specifying various

profiles and parameters agree on the semantics of those parameters that impact negotiation. In our previous work [16], we have specified a generic data specification model identifying the parameters specified by various actors impacting the QoS negotiation process, including user and service profiles.

Based on messages received during QoS (re)negotiation, the *Signaling logic function* decides when to invoke the *Matching Process* and the *Optimization Process* (Fig. 3). The matching and optimization algorithms are implemented independent of a particular service. With different services requiring different levels of QoS support (e.g., a mission critical telemedicine service vs. streaming music clips), the *QMO strategy configuration* function may decide on the complexity of the matching and optimization processes to be used. In general, the processes should be able to handle profiles/parameters of arbitrary complexity. For example, a particular user may have charging requirements specified in his/her profile that need to be considered while another user does not. Furthermore, providing advanced QoS support will not always be necessary (or desirable) due to increased processing complexity and signaling overhead.

Input data can be collected from entities for a given session, or can be collected once and stored by the Q-MOF to be used for multiple sessions. Data that needs to be stored is passed to either the *Input data storage* (stores data that is not session specific, such as service costs, general operator policy, etc.) or the *Temporary storage* (stores session specific data which is erased when a session is ended). During the course of service execution, the Q-MOF may receive various adaptation-triggering events indicating the need for renegotiation and adaptation. Based on user preferences or service requirements, the Q-MOF can also subscribe to various contextual events that may lead to adaptation, such as subscribing to user location information.

3.1 QoS Matching Process

The aim of the Matching Process is to parse collected input data and match service requirements against the capabilities/requirements of service users, and any additional imposed constraints (e.g., imposed by network policy) in order to determine feasible service parameters. The process is invoked through the *Matching Context Identification Function* (MCIF) which is responsible for determining a set of matching parameters p_1, \dots, p_k for a given service to be used as a basis for the actual matching process conducted by the *Matching Engine* (ME). Matching parameters represent those service requirements that need to be checked to make sure they can be met. An example of a matching parameter set would be $\{min_bandwidth_downlink, min_bandwidth_uplink, media_required, codecs\}$. The selection of matching parameters is based on requirements specified in the Service Profile and will depend on the type of service being established (i.e., different services may specify requirements in terms of different parameters). Parsing of the Service Profile to identify matching parameters may be done once for each new service rather than for each incoming user request.

Based on the identified parameter set p_1, \dots, p_k , the ME conducts the actual matching process (Figure 4). The ME executes matching rules for each parameter

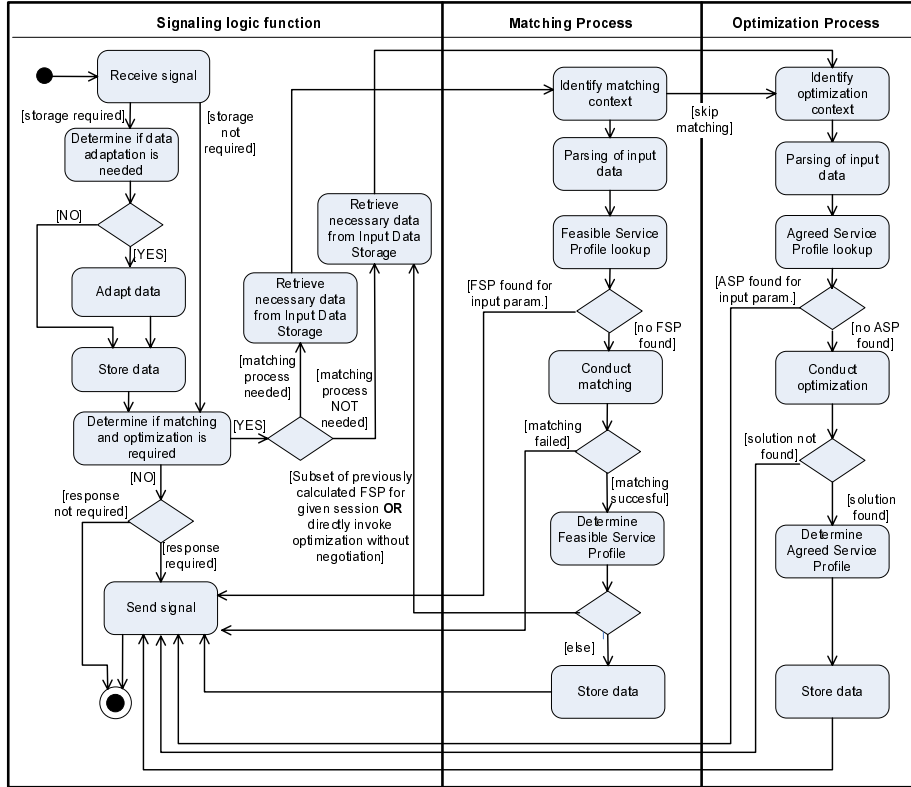


Fig. 3. Q-MOF activity diagram: handling a received signal

included in the matching parameter set. The rules are of the form: *if* $\langle p_i == p \rangle$ *then* $\langle action \rangle$. For example, *if* $\langle p_i == min_bandwidth_uplink \rangle$ *then* $\langle check\ if\ user\ capabilities\ and\ network\ constraints\ are\ greater\ than\ min_bandwidth_uplink \rangle$.

After completing the matching, the ME will create an FSP that specifies all feasible service versions and operating parameters determined. After entities have agreed to the FSP, the FSP is passed on to the Q-MOF Optimization Process. If matching results indicate that requirements cannot be met, the Q-MOF will signal that no FSP can be found.

We note that in the case of an adaptation-triggering event received during service execution, the ME will execute matching rules only for updated parameters rather than repeating the entire process. The ME will compare new and old matching results to see if new results are a subset of old results. If they are a subset, there would be no need to repeat negotiation, and the Optimization Process may be directly invoked. If new matching results are not a subset of old results, a new FSP is determined, stored, and signaled to involved entities.

In networks with a large number of session requests, it may be too time-consuming to conduct the matching process for each new request. Assuming that input parameters to the matching process may often times be the same as

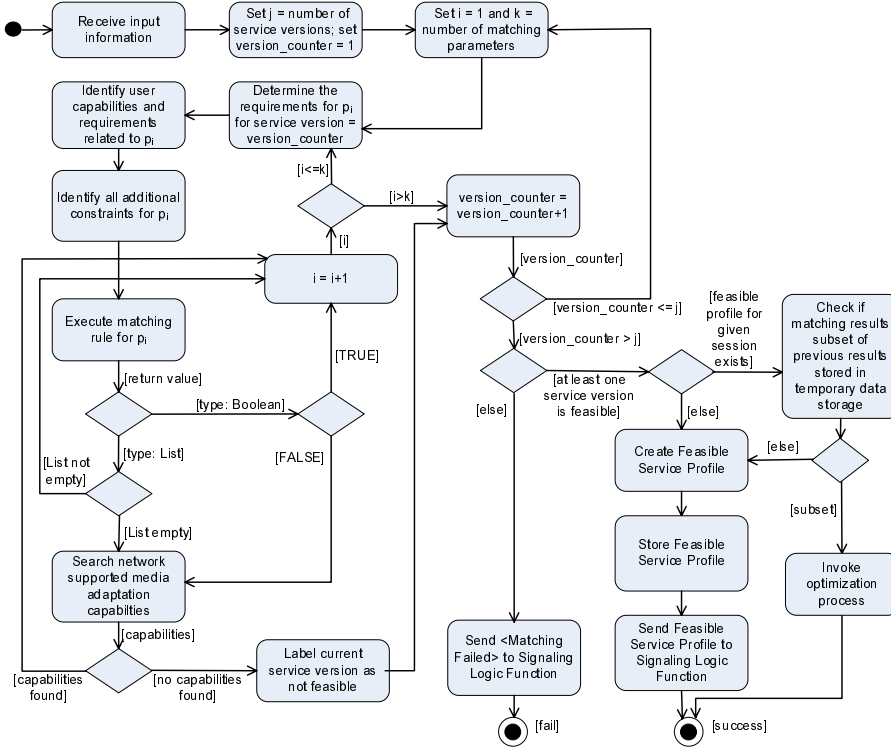


Fig. 4. Activity diagram for Matching Engine

previously signaled input parameters corresponding to an earlier session request (i.e., multiple users will often times have the same or similar user profiles in terms of terminal/access network capabilities, preferences; operator policy may be the same across multiple users), we propose for the Q-MOF to store calculated FSPs for representative sets of input parameters which may be used for subsequent requests. With each new request for a given service, input data is parsed based on the matching parameter set for that service to determine if a request with the same relevant parameter values (or ranges of values) has already been processed. A hashing process may be used to encode parsed input parameters and enable efficient FSP profile lookup. If a previously calculated FSP already exists for the given input parameters, there is no need to invoke the ME and the previously calculated FSP is signaled.

3.2 QoS Optimization Process

The Optimization Process receives as input the negotiated FSP and produces the ASP. As in the case of matching, the Optimization Process can check whether a previously calculated ASP already exists for an equivalent negotiated FSP.

Hence, in the case of a large number of service requests, previously calculated optimization results may be reused.

Calculation of the ASP is based on the service profile specifying a mapping between a point in an Operating Space O whose dimensions constitute service-level parameters that are being negotiated (e.g., codec, frame rate); a Resource Space R indicating the (network and system) resources required to support that operating point; and a Utility Space U , which may consist of multiple dimensions representing both qualitative and quantitative quality measures. Such a mapping has been discussed previously in [17] and has been adopted in the MPEG-21 Digital Item Adaptation (DIA) standard [4] to be used for meta-data driven QoS adaptation. We adopt these mechanisms and apply them in the broader context of service-level (re)negotiation.

The *Optimization Context Identification Function* (OCIF) is responsible for determining the optimization objective, collecting necessary input data, and invoking the *Optimization Engine* (OE). The OE will then run the actual optimization algorithm(s), and determine the ASP based on calculation results. In determining the optimization objective, the OCIF can conclude that the problem to be solved is either a single-objective or multi-objective problem. As feasible media operating points may be mapped to multiple utility dimensions, the objective of maximizing utility may be considered a multi-objective problem. In this paper we focus on single-objective optimization.

The problem to be solved is the optimal distribution of constrained resources among multiple media flows. We assume a finite number of operating points per media flow. The goal is to choose exactly one operating point per media flow. For simplification purposes, we assume a one dimensional utility space corresponding to user perceived value. We consider total utility to be the weighted sum of the individual media flow utilities. Different user preferences and context information may be used for assigning weight factors (WF) and formulating the optimization problem. For preferences that are service or media flow specific, the Service Profile may provide the rules for assigning WFs (e.g., if $\langle pref == \text{"audio more important than video"} \rangle$ then $\langle audio_WF = 1 \text{ and } video_WF = 0.5 \rangle$).

The general problem may be formulated as follows. We assume n different media flows, of which flows $1, \dots, h$ are in the downlink direction and flows $h + 1, \dots, n$ are in the uplink direction. The i th flow has p_i operating points. Required resources \mathbf{r} for operating point j and media flow i are denoted as $\mathbf{r}_{ij} = (r_{ij1}, \dots, r_{ijq}, \dots, r_{ijk})$.

Assuming q different QoS classes, we specify r_{ij1}, \dots, r_{ijq} as corresponding to bandwidth assigned within different QoS classes (differing in delay, loss, jitter, and bandwidth guarantees). For a single operating point, we assume only one of the values r_{ij1}, \dots, r_{ijq} to be greater than zero, while all others are equal to zero. This is because only one QoS class is chosen per operating point for a media flow. We define variables $B_{downlink}$ and B_{uplink} to denote maximum available downlink and uplink bandwidth respectively, determined by user terminal capabilities and access network. They represent the resource constraints for r_{ij1}, \dots, r_{ijq} . Additional resource constraints (other than bandwidth) related

to the consumption of a particular resource across all media flows are expressed as $\mathbf{R} = (R_{q+1}, \dots, R_k)$ (e.g., cost for all flows must be less than a specified amount). Resource constraints may also be added regarding resource consumption per flow and expressed as $\mathbf{R}_i = (R_{i1}, \dots, R_{ik})$ (e.g., cost for a media flow must be less than a specified amount). The utility value for operating point j and media flow i is denoted as $u_i(\mathbf{r}_{ij})$. Weight factors w_i are assigned to utility values to indicate the relative importance of media flows. We include 0-1 variables x_{ij} to make sure that exactly one operating point is chosen per media flow.

Below we show formulation for the case of multi-flow resource constrained utility maximization. The problem is formulated as a multi-choice multi-dimension 0-1 knapsack problem (MMKP). Given a set of media flows, each with a number of specified operating points, the best operating point per flow is selected.

$$\max \sum_{i=1}^n \sum_{j=1}^{p_i} w_i x_{ij} u_i(\mathbf{r}_{ij}) \quad (1)$$

such that

$$\sum_{i=1}^k \sum_{j=1}^{p_i} \sum_{y=1}^q x_{ij} r_{ijy} \leq B_{downlink} \quad (2)$$

$$\sum_{i=k+1}^n \sum_{j=1}^{p_i} \sum_{y=1}^q x_{ij} r_{ijy} \leq B_{uplink} \quad (3)$$

$$\sum_{i=1}^n \sum_{j=1}^{p_i} x_{ij} r_{ijy} \leq R_y, \quad y = q + 1, \dots, k \quad (4)$$

$$\sum_{j=1}^{p_i} x_{ij} r_{ijy} \leq R_{iy}, \quad y = 1, \dots, k; \quad i = 1, \dots, n \quad (5)$$

$$\sum_{j=1}^{p_i} x_{ij} = 1, \quad i = 1, \dots, n \quad (6)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, n; \quad j = 1, \dots, p_i \quad (7)$$

The Q-MOF may implement different optimal solution algorithms and heuristics (for finding near optimal solutions for cases with a large number of variables) for solving the formulated problem (e.g., such as those proposed in [10]).

While we have focused on generic problem formulation, we note that a different approach to formulating the presented optimization problem (out of scope for this paper) may be to use tools specified as part of the MPEG DIA standard, namely Usage Environment Description tools, Universal Constraints Description tool, and AdaptationQoS tools. A decision-taking framework based on these tools and supporting media adaptation is described in [11].

4 Case Study

We illustrate Q-MOF functionality using a prototype **Audio/Video Call (AVC)** service allowing two end users to engage in an audio/video call. The service is implemented in a laboratory IMS testbed (details on testbed can be found in [14]). The Q-MOF has been added as a SIP AS along the E2E SIP signaling path and is invoked by the IMS Serving Call Session Control Function (S-CSCF). User and service profiles are specified using XML and included in SIP signaling. Furthermore, we have defined a policy file composed of IMS operator policy and access network policy to be used as input for QoS negotiation.

The service is offered in two versions: (1) *AVC 1* with conversational audio and video; and (2) *AVC 2* with only conversational audio. We assume the uplink direction as being from the call initiating user (User A) to the terminating user (User B). The AVC service matching parameter set (p_1, \dots, p_{12}) is shown in Table 1. The AVC service specifies both network and processing (resolution) resource requirements.

We further illustrate a case of invoking the Optimization Process when the chosen service version is *AVC 1*. Operating parameters, resource vectors and corresponding utilities are specified in the service profile and summarized in Table 2 (parameters and resource vectors for uplink flows are identical and ordered the same as for downlink flows). For illustration purposes, we assume a specification

Table 1. Matching parameter set for AVC service

Matching parameters for AVC service	Service requirements (version <i>AVC 1</i>)	User A capabilities	User B capabilities	Operator constraints
$p_1 = \text{media_components}$	audio, video	audio, video, data, image, model	audio, video, data, image, model	User A and B: audio, video, data, image, text
$p_2 = \text{codecs}$	audio : mpeg, gsm; video : mpeg, h263	audio: mpeg, pcm, gsm; video: mpeg, mjpeg, h263	audio: mpeg, pcm, gsm; video: mpeg, mjpeg, h263	notAllowed : audio, G729, audio G723
$p_3 = \text{minBandwidth_DL}$	46	1200	1300	1400
$p_4 = \text{maxDelay_DL}$	150	150	150	N/A
$p_5 = \text{maxJitter_DL}$	10	N/A	N/A	N/A
$p_6 = \text{maxLoss_DL}$	1	N/A	N/A	N/A
$p_7 = \text{minBandwidth_UL}$	46	800	1000	1400
$p_8 = \text{maxDelay_UL}$	150	150	150	N/A
$p_9 = \text{maxJitter_UL}$	10	N/A	N/A	N/A
$p_{10} = \text{maxLoss_UL}$	1	N/A	N/A	N/A
$p_{11} = \text{resolution_local}$	176x144	1024x768	N/A	N/A
$p_{12} = \text{resolution_remote}$	176x144	N/A	1024x768	N/A

Table 2. Oper. parameters, resource vectors and utilities for AVC DL flows

Media flow	Operating parameters	Resource vectors	Bandw. [kbps] r_{ij1}	Cost [monetary_unit/s] r_{ij2}	Utility $u_i(\mathbf{r}_{ij})$
audio DL	codec: GSM, sample rate: 8000, bits per/samp: 8	r_{11}	21	210	0.5
	codec: MPEG, sample rate: 22050, bits per/samp: 16	r_{12}	34	340	0.80
	codec: MPEG, sample rate: 44100, bits per/samp: 16	r_{13}	64	640	1.00
video DL	codec: H263, res: 176x144, framerate: 5	r_{21}	25	900	0.20
	codec: H263, res: 176x144, framerate: 15	r_{22}	90	900	0.4
	codec: MJPEG, res: 176x144, framerate: 5	r_{23}	370	3700	0.5
	codec: MJPEG, res: 176x144, framerate: 10	r_{24}	400	4000	0.6
	codec: MJPEG, res: 176x144, framerate: 15	r_{25}	781	7810	0.7
	codec: MJPEG, res: 352x288, framerate: 5	r_{26}	1015	10150	0.8
	codec: MJPEG, res: 352x288, framerate: 15	r_{27}	1400	14000	0.9
	codec: MJPEG, res: 352x288, framerate: 30	r_{28}	2000	20000	1

of utility in the range of $[0,1]$. A different approach may be to specify utility in terms of a Mean Opinion Score (MOS) on a scale from 1 to 5, often used to provide a numerical indication of subjective user perceived quality.

The number of feasible operating points is as follows: audio downlink (DL) ($p_1 = 3$), video DL ($p_2 = 8$), audio uplink (UL) ($p_3 = 3$), and video UL ($p_4 = 8$). We assume that cost is based on amount of traffic and assigned QoS class. We therefore calculate cost as $bandwidth_class_q$ [bit/s] \cdot $price_class_q$ [monetary_unit/b]. For the conversational audio/video QoS, we assume a hypothetical price of 10 [monetary_unit/bit].

For the optimization problem we assume user preferences which indicate that audio is preferred over video and set an audio WF to 1 and video WF to 0.7. The following constraints are determined based on user profiles: maximum DL bandwidth 1200 kbps, maximum UL bandwidth 800 kbps, and maximum cost 20000 [monetary_unit/s] (corresponds to User A's budget, assuming that User A will cover all costs). The result of the optimization is an ASP including an MDP. The MDP is composed of one optimal and four suboptimal service configurations, as shown in Table 3. We combine certain configurations and convert calculated bandwidth values to bandwidth ranges for media flows with continuous utility along those ranges (in this case video UL and video DL).

Table 3. Resource configurations for the AVC MDP

Resource configuration	Audio DL	Video DL	Audio UL	Video UL	Objective value
1 (optimal)	r_{13}	r_{26}	r_{33}	$[r_{43}, r_{44}]$	$[2.909, 2.980]$
2	r_{13}	$[r_{24}, r_{25}]$	r_{33}	r_{44}	$[2.840, 2.909]$
3	r_{13}	r_{26}	r_{33}	r_{42}	2.840
4	r_{13}	r_{25}	r_{33}	r_{43}	2.839
5	r_{13}	r_{23}	r_{33}	r_{44}	2.77

Table 4. Processing times for Matching Process

Matching results	Service establishment [s]	Update) [s]
Feasible service versions	0.531	0.683
No feasible service versions	0.245	0.670

For experimentation purposes, the Q-MOF was run on a Pentium IV PC with 2.4 GHz and 512 MB RAM. Signaled profiles are parsed using a Simple API for XML (SAX) parser. Measurements showing the Q-MOF matching processing time (including profile parsing) are given in Table 4 and correspond to: (a) initial service establishment, and (b) an update received during an active service indicating a change in the value of one matching parameter, e.g., decrease in user bandwidth. Both cases correspond to the situation when the ME is invoked. Results show that in the case when the ME finds no feasible service versions, processing time is significantly reduced as there is no FSP being created. Processing time is greater for the update scenario than for initial service establishment due to the comparison with previously found matching results for the given session.

Results showing the Q-MOF optimization processing times including MDP calculation are given in Table 5. We use the GLPK (GNU Linear Programming Kit) [20] simplex-based solver and MIP problem solver to search for an optimal solution. Measurements are conducted for different numbers of active media components and operating points. We added additional hypothetical active media streams to the FSP in order to see the effects on processing time. The number

Table 5. Processing times for Optimization Process

No. of active media components (n)	No. of oper. points $\sum_{i=1}^n p_i$	No. of feasible combinations $\prod_{i=1}^n p_i$	No. of found cfigs.	No. of chosen MDP cfigs.	Proc. time (single execution) [s]	Total proc. time [s]
2	6	9	7	7	0.016	0.264
4	22	576	9	7	0.016	0.383
4	44	9216	7	7	0.016	0.430
6	76	23592	16	7	0.031	1.933

of found service configurations corresponds to the number of times that the optimization algorithm was run. Found solutions where neither the total UL nor DL bandwidth decreased as compared to a previously found solution were not included in the MDP. Those solutions that were included in specifying the MDP are referred to as chosen MDP configurations. The total processing time for the optimization process includes parsing of the FSP and is significantly greater than processing time for the optimization algorithm (refers to single algorithm execution in searching for the optimal MDP configuration assuming profiles have been preciously parsed). It is clear that execution times greatly increase with the number of active media components. For cases in which a greater number of parallel streams are present, heuristics for providing fast and near optimal solutions may be used [10]. Another option is for the Q-MOF to conduct optimization procedures offline and store solutions offering certain combinations of feasible service parameters for groups of end users with common characteristics.

5 Discussion and Conclusions

The functionality proposed in this paper has focused on service-level QNA mechanisms for the NGN architecture. The novel contribution is the model of a Q-MOF based on matching restrictive user, service, and network requirements, policies and constraints, with the goal of maximizing utility. The proposed approach is applicable for the NGN architecture as discussed in Section 2. Furthermore, the Q-MOF may serve to enhance network-initiated QoS control mechanisms, such as those adopted in the 3GPP Release 8 Evolved Packet System [6]. Considering an IMS-based network architecture, the Q-MOF may be introduced at the application signaling level to determine optimal service resource and authorization requests which are then passed to the network.

Regarding scalability issues, it is clear that for a large number of users, running the matching and optimization procedures separately for each service session may be too time consuming and costly. We have therefore proposed for the Q-MOF to store service configuration solutions calculated in advance for particular combinations of constraints (as opposed to per user/per service request basis). While the presented case study serves to illustrate Q-MOF functionality, future work will focus on testing scalability of the proposed solution in cases involving a large number of parallel user sessions. Future research will explore the possibility for categorization of users and service requests in making domain-wide optimization decisions. Furthermore, research will focus on QoS negotiation models in cases involving multiple providers in service delivery.

References

1. ITU-T Recommendation Y.2012. Functional requirements and architecture of the NGN release 1 (2006)
2. ITU-T Recommendation E.860. Framework of a service level agreement (2002)
3. 3GPP TS 23.228: IP Multimedia Subsystem (IMS); Stage 2, Release 8 (2008)

4. Information Technology - Multimedia framework (MPEG-21) - Part 7: Digital Item Adaptation, ISO/IEC 21000-7 (2004)
5. Boula, L., Koumaras, H., Kourtis, A.: An Enhanced IMS Architecture Featuring Cross-Layer Monitoring and Adaptation Mechanisms. ICAS, Spain (2009)
6. Ekstrom, H.: QoS control in the 3GPP Evolved Packet System. *IEEE Comm. Magazine* 47(2), 76–83 (2009)
7. Guenkova-Luy, T., Kassler, A.J., Mandato, D.: End-to-End Quality-of-Service Coordination for Mobile Multimedia Applications. *IEEE J. Sel. Areas Commun.* 22(5), 889–903 (2004)
8. Handley, H., Jacobson, V.: SDP: Session Description Protocol. IETF RFC 2327 (1998)
9. Houssos, N., et al.: Advanced Adaptability and Profile Management Framework for the Support of Flexible Service Provision. *IEEE Wireless Communications*, 52–61 (August 2003)
10. Khan, S.: Quality Adaptation in a Multisession Multimedia System: Model, Algorithms and Architecture. PhD Thesis, Univ. of Victoria (1998)
11. Mukherjee, D., Delfosse, E., Kim, J.-G., Wang, Y.: Optimal Adaptation Decision-Taking for Terminal and Network Quality of Service. *IEEE Trans. on Multimedia* 7(3), 454–462 (2005)
12. Ozelebi, T., Radovanovic, R., Chaudron, M.: Enhancing End-to-End QoS for Multimedia Streaming in IMS-Based Networks. In: ICSNC, pp. 48–53 (2007)
13. Rosenberg, J., et al.: SIP: Session Initiation Protocol. IETF RFC 3261 (2002)
14. Skorin-Kapov, L., Mosmondor, M., Dobrijevic, O., Matijasevic, M.: Application-level QoS Negotiation and Signaling for Advanced Multimedia Services in the IMS. *IEEE Comm. Magazine* 45(7), 108–116 (2007)
15. Skorin-Kapov, L., Matijasevic, M.: A QoS Negotiation and Adaptation Framework for Multimedia Services in NGN. In: Proceedings of ConTEL 2009, Zagreb, Croatia, pp. 249–256 (2009)
16. Skorin-Kapov, L., Matijasevic, M.: A Data Specification Model for Multimedia QoS Negotiation. In: Proc. ACM Mobimedia (2007)
17. Wang, Y., Kim, J.-G., Chang, S.-F., Kim, H.-M.: Utility-Based Video Adaptation for Universal Multimedia Access (UMA) and Content-Based Utility Function Prediction for Real-Time Video Transcoding. *IEEE Trans. on Multimedia* 9(2), 213–220 (2007)
18. IST DAIDALOS - EU FP6, <http://www.ist-daidalos.org/>
19. IST ENTHRONE - EU FP6, <http://www.ist-enthrone.org/>
20. GLPK (GNU Linear Programming Kit), <http://www.gnu.org/software/glpk/glpk.html#T0Cintroduction> (last accessed November 10, 2008)