Enabling self-adaptive QoE/QoS control

Solange Rito Lima, Paulo Carvalho
Centro Algoritmi, University of Minho, Department of Informatics
4710–057 Braga, Portugal
e-mail:{solange,pmc}@di.uminho.pt

***Abstract*—Handling quality requirements of multimedia services and the expectations of end-users regarding the perceived service quality is currently a major issue for service providers in order to sustain service diversity and improve competitiveness. In this context, this paper presents ongoing work toward a service-oriented architecture for QoE/QoS evaluation and control, which can be deployed to assist the provision of multi-constrained services. Considering the users' QoE perspective and the negotiated service levels, the architecture lays on per service class online monitoring to assist self-adaptive control of multimedia flows entering the network. To perform online monitoring, a distributed and versatile QoS monitoring tool oriented to multiservice networks is proposed. Preliminary results shows that the presented control strategy is effective in providing consistent quality levels to heterogeneous services.**

I. Introduction

The integration of new value-added network multimedia services, such as IP telephony, video on demand and IPTV in a common infrastructure, increase service providers concern and pressure regarding the provision of appropriate service quality levels. Satisfying the diversity of user demands and quality expectations in a competitive and cost-effective manner are main motivations for deploying efficient and service-oriented architectures for quality of experience (QoE) and quality of service (QoS) management.

This paper is focused on proposing an architecture for QoE/QoS assessment and control for next generation networks. The architecture design is oriented to assure a consistent service delivery through self-adaptive quality control inside a network. This control is achieved articulating two main components: QoS and service level specification (SLS) monitoring, and service-oriented admission control (AC). In this context, we propose: (i) a distributed service-oriented monitoring model including essentially software-based measurements points (MPs) and a QoS server cooperating to assess each service class status and performance; (ii) a distributed service-oriented AC model based on edge-to-edge on-line QoS and SLS monitoring to perform lightweight AC in multiservice networks. To sustain the real applicability of the proposed QoE/QoS architecture, a proof-of-concept illustrating the model's self-adaptive ability to control distinct service levels is provided.

This paper is organized as follows: related work is discussed in Section II; the QoE/QoS management architecture and its main components is presented in Section III; its proof-of-concept is provided in Section IV; and the conclusions are presented in Section V.

II. Related work

This section discusses relevant work related to the main components of the QoE/QoS management architecture.

Besides IETF IPPM and ITU-T efforts in defining relevant metrics and measurement methodologies for IP networks, recent standardization activities have extended former work with proposals for the evaluation and management of QoS and QoE associated with speech and video [1]. In particular, video services (e.g. IPTV), expected to be the most representative and growing service in the forthcoming years [2], have deserved special attention regarding the definition and quantification of QoE and QoS metrics. The standard ITU-T G.1080 [3] identifies QoE requirements following a multilayer approach. From an application and transport layer perspective, QoE parameters are related with objective, quantifiable QoS parameters (e.g. stream bit rate, latency, jitter and loss ratio) and video coding technologies. In addition, ITU-T G.1081 [4] is focused on monitoring IPTV quality, identifying the location of measurement points, quality parameters and measurement methods. This will allow service providers or network operators monitoring the performance of service distribution to end customers, regarding aspects such as resources utilization and optimization, and service reliability.

Currently, there is a vast range of projects and tools covering multiple network monitoring aspects. RIPE NCC project [5], as an operational network-monitoring platform, is a good example of how to provide active measurements for ISPs, but it does not support service differentiation. Other network monitoring solutions mostly evaluate the network performance from a service monolithic perspective (see [6] for a wide variety of tools). These solutions are limitative within multiservice networks context, where the main concerns regarding monitoring are related to a scalable and encompassing evaluation of differentiated QoS delivery. This reasoning and analysis grounds the motivation for developing a new service-oriented monitoring tool (*QMon*).

Regarding AC, usual approaches for multiservice IP networks can be categorized as centralized (e.g. based on bandwidth brokers) or distributed, parameter and/or measurement-based, differing on the type of services being supported. Measurement-based AC (MBAC) and end-to-end MBAC (EMBAC) solutions have deserved special attention, as they lead to reduced control information and overhead, but eventually to QoS degradation. These AC approaches are surveyed and compared in [7].

III. An architecture for QoE/QoS control

The definition and configuration of network services cannot be decoupled from the high-level quality and performance requirements of users applications. Depending on the type of services and applications, the users have, and may express, their perception on how those services are being experienced. These levels of quality have to be taken into account when service providers need to specify, configure and monitor network services and underlying resources.

Figure 1 provides a high-level vision on QoE/QoS management, integrating both the user and the service provider perspectives. As shown, the QoE/QoS management life-cycle starts with the end-user quality perspective for multimedia services, expressed by a set of qualitative and/or quantitative QoE/QoS requirements. These requirements are usually defined through SLSs celebrated between the parties. In practice, to allow a consistent control of QoS inside a network domain, both QoE/QoS descriptions should be mapped to quantifiable values with more or less relaxed thresholds, to be effectively controlled during network operation. In this way, the service levels defined and negotiated between customers and providers, will provide guidelines for configuring service classes and for controlling the respective quality levels. Service control, taken at two levels - QoS and SLSs control - is achieved articulating QoS/SLSs monitoring and service-oriented AC. The QoE/QoS management life-cycle is closed with performance assessment of multimedia services, resorting to monitoring feedback. This last step includes SLS auditing, online service levels' visualization and reporting, to assist both users and service provider activities.

Figure 2 illustrates how the main model components interoperate for achieving a self-adaptive service management. Taking the specified requirements for each service class and the corresponding QoS objectives, the network is systematically monitored on a per service basis in order to trigger proper traffic control actions. These control actions include deciding on the admission of incoming multimedia flows, according to the measured QoS levels of the service classes and corresponding thresholds.

The following sections focus on the inner control loop of the diagram represented in Figure 2.

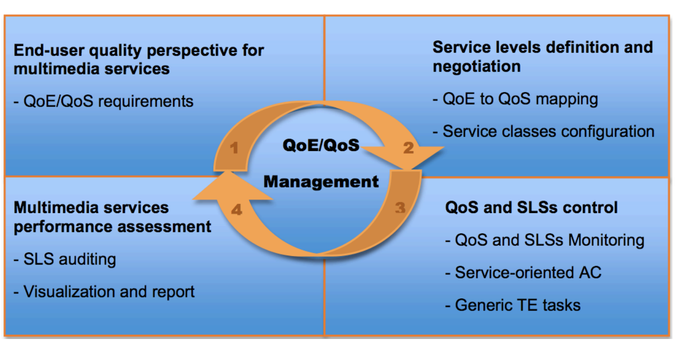

Fig. 1. QoE/QoS management life-cycle

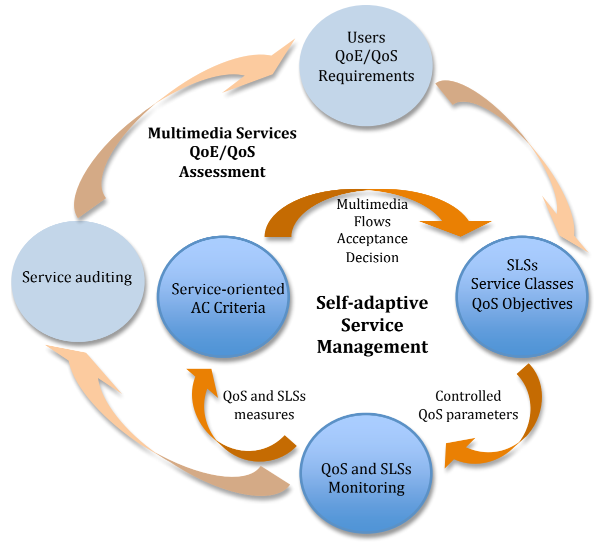

Fig. 2. Components for self-adaptive service management

A. Service-oriented Admission Control

The AC model presented in [8] resorts to edge-to-edge on-line monitoring to obtain feedback of each service class SC_i performance so that proper AC decisions can be made. For a dynamic control of traffic entering a network domain, the underlying AC rules control both QoS levels in the domain and the sharing of active SLS between domains, according to rules defined in Table I. When measured QoS levels are not in conformance with the SC_i QoS objectives, new flows are not accepted until the class recovers and restores QoS target values; the same occurs when downstream SLSs utilisation is above the negotiated capacity. Both equations assume safety margins to deal with optimistic measurements.

As regards the *end-to-end case*, both AC and available service computation are seen as a repetitive and cumulative process, performed at each domain's ingress nodes.

B. QoS and SLS monitoring - QMon tool

The main objective of QMon, as a flexible, open-source software monitoring tool, is to provide systematic QoS monitoring in multiservice networks. This tool allows to: (i) keep track of ongoing QoS and network performance levels; (ii) verify SLS compliance; and (iii) provide feedback to traffic control mechanisms and trigger network recovery procedures.

QMon encompasses two main elements: the QBoxes and the QServer. QBoxes are MPs strategically distributed in the network, typically at its boundaries (see [4] for MPs location regarding IPTV service monitoring), and their main task is to exchange probing traffic to compute the QoS metrics of each service class SC_i. The QServer gathers all measurements and QBoxes' configurations into a central database. Collected data remains available for subsequent analysis and support of management and traffic engineering actions over distinct time scales. In medium/short term the measures remain available in the QBoxes to support distributed traffic engineering, avoiding

TABLE I
CONTROL RULES SUMMARY

TYPE OF RULE	DESCRIPTION
SLS Rate Control Rules	Verify downstream SLSs utilization
$\tilde{R}_{i,(*,E_m)} + r_j \leq \beta_{i,E_m} R_{i,E_m}$	$\tilde{R}_{i,(*,E_m)}$ - current measured rate of flows using downstream SLS_{i,E_m}, considering all ingress-to-E_m estimated rates of flows going through egress E_m; r_j - rate of the new flow F_j; $0 < \beta_{i,E_m} \leq 1$ - service-dependent safety margin for the rate R_{i,E_m} defined in SLS_{i,E_m}.
QoS Control Rules	Verify the conformance of QoS levels in the domain
$\forall (P_{i,p}, \beta_{i,p}) \in P_{SC_i} : \tilde{P}_{i,p} \leq T_{i,p}$	$\tilde{P}_{i,p}$ - ingress-to-egress measured QoS parameter; $\beta_{i,p}$ - corresponding safety margin; $T_{i,p}$ - parameter's upper bound or threshold, given by $T_{i,p} = \beta_{i,p} P_{i,p}$, used to set the acceptance status in Δt_i.
End-to-end Control Rules	Cumulative computation of e2e QoS
$\forall P_{j,p} \in P_{F_j}$: $(\mathtt{op}_1\ (P^{acc^-}_{j,p}, P_{i,p}))\ \mathtt{op}_2\ (\gamma_{j,p} P_{j,p})$	$P_{j,p}$ - flow QoS parameter, allowing a tolerance factor $\gamma_{j,p}$; $P^{acc^-}_{j,p}$ - parameter's cumulative value when crossing upstream domains; $P_{i,p}$ - corresponding target value in present domain.

the functional dependence and congestion of a single central entity. QBoxes and QServer are software-based elements, more specifically Java applications. QMon is, therefore, highly portable, easy to deploy and cost-effective.

The status of all existing QBoxes and their configuration is maintained in QServer. This allows changing data on-the-fly by sending them update messages. QServer is also responsible for notifying the QBoxes when a new QBox becomes active or inactive, or when new service measurement policies need to be deployed, without disruption of the monitoring service.

The communication among the QServer and the QBoxes is accomplished through the primitives illustrated in Figure 3.

A. Initializing a QBox comprises the following phases: (1) Initially, the new QBox sends its identification to the QServer for registration, enabling the QBox to be included in the measuring process. After validation, the QServer sends back an OK message and notifies the other QBoxes that a new box is in place (B); (2) The QServer updates QBox data based on two distinct sequences of messages: one informing the characteristics of the defined service classes; the other providing each QBox IP address and identification. Then, the new QBox is ready to send and receive probing traffic to and from all QBoxes. The characteristics of probing traffic and the metrics to evaluate are service-dependent, according to a configuration previously received; (3) Every time a sample of probing traffic is fully received by a QBox, the metrics of the corresponding service class are calculated and sent to the QServer, which stores them in a database.
B. After a new QBox is registered, the QServer notifies all other active QBoxes of its existence, using the already open sessions. Thus, adding new QBoxes to the platform dynamically does not interrupt ongoing measurements;
C. When a QBox shuts down, the existing session between this QBox and the QServer is closed. The QServer notifies this occurrence to other QBoxes in order to stop the measurement flow to the offline QBox;
D. Whenever a new service class is defined or updated in the database, its new settings are also sent to all the involved QBoxes. Thus, probing to new service classes may start without interrupting the measurement procedures of the remaining classes;
E. Similarly, whenever a service class is removed from the database, the QServer notifies all the involved QBoxes so that the obsolete class can be removed from their local information structure, Once again, measurements in the remaining service classes are not disrupted.

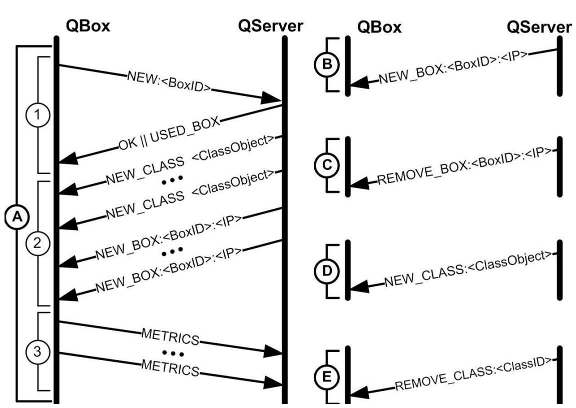

Fig. 3. QBox/QServer Communication Primitives

To access the measurement results, a web front-end has been developed. The front-end allows rendering different graphs illustrating, for all service classes, the QoS metrics behaviour between QBoxes over distinct, configurable time scales. To interact with the database a generic API has been developed, which allows performing queries and statements to MySQL or Oracle 10g databases.

IV. PROOF-OF-CONCEPT

At the present stage, QMon prototype is fully operational, stable and tested in a local testbed. A medium-term aim is to extend this testing to PlanetLab in order to assess and validate the tool over a large-scale network environment. To test the service-oriented AC component and its self-adaptability, the developed prototype considers three service classes configured according [9] guidelines. Table II summarizes the service classes implemented and AC parameters. As shown, the parameterization of the AC rules is service-dependent and larger safety margins β^+_{i,E_m} and tighter thresholds $T_{i,p}$ are defined for more demanding classes. The AC thresholds $T_{i,p}$ considers domain's characteristics and QoE/QoS upper bounds for common applications and services [1], [3], [10].

The network domain consists of ingress routers I_1, I_2, a multiclass network core and an edge router E_1. I_2 is used to inject concurrent or cross traffic (referred as CT-I2), allowing

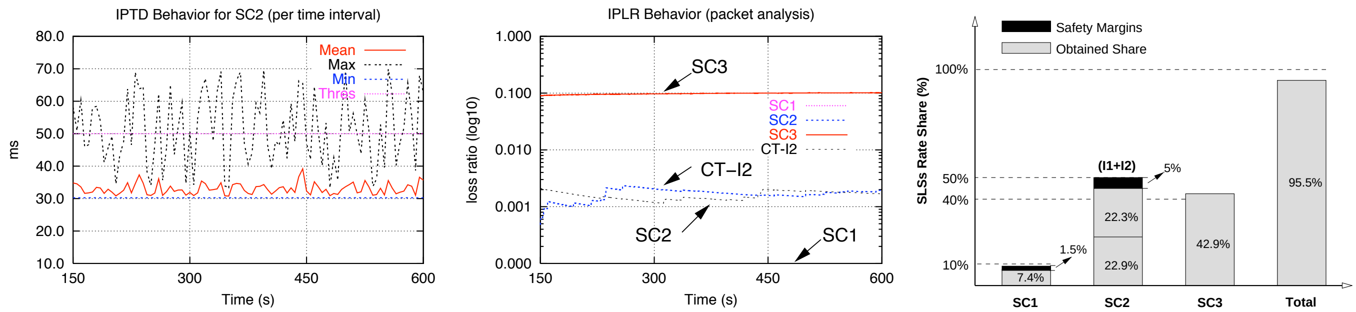

Fig. 4. IPTD for SC2; IPLR and rate share for all classes;

TABLE II
SERVICE CLASSES SC_i

SC_i	$R^+_{i,Em}$	$\beta^+_{i,Em}$	$P_{i,p}$	$T_{i,p}$	Traffic Type
SC1	10% BW share	0.85	IPTD ipdv IPLR	35ms 1ms 10^{-4}	VoIP Real-Time Interactive Broadcast Video
SC2	50% BW share	0.90	IPTD IPLR	50ms 10^{-3}	Multimedia streaming Low-Latency Data
SC3	40%	1.0	IPLR	10^{-1}	High-Throughput Data

to evaluate concurrency effects on distributed AC and assess the impact of cross traffic on the model performance.

A. Assessing metrics control effectiveness

The results show that (see Figure 4):

i) SC1 is well controlled presenting a stable QoS behavior. IPTD is kept almost constant throughout the simulation period. The mean ipdv assumes a low value as a result of small variations, bounded by a well-defined maximum and minimum value. IPLR, around $0.9\text{x}10^{-4}$, is below the defined threshold;

ii) for SC2, although the mean IPTD is well-bounded, in some time intervals, the maximum IPTD exceeds the defined thresholds. The statistical analysis of the involved time series shows that the number of packets exceeding the QoS thresholds is very small. Recall that a QoS threshold violation does not necessarily imply a QoS violation, as the concept of threshold comprises a safety margin to the QoS parameter target value;

iii) SC3 IPLR evolution tends to the defined IPLR threshold of 10^{-1}. For SC2 traffic, IPLR has a less continuous behavior as it results from occasional loss events, converging to the defined threshold of 10^{-3};

iv) the total utilization is kept high, and each class rate share is well accomplished.

When examining the way AC rules (see Table I) determine the control behavior discussed above, the following is identified: (i) SC1 flows are controlled essentially by the SLS rate control rule as a result of a stable QoS behavior associated with this high priority class; (ii) AC for SC2 flows is triggered by SLS and/or QoS control rules; (iii) SC3 flows are controlled by the QoS control rule; (iv) IPLR violations have a predominant role in setting the AC status to a rejection mode in the QoS control rule. Knowing which AC rule has more influence on the AC decision process can also bring relevant information and directions for improving service configuration and provisioning both intra and interdomain.

V. CONCLUSIONS

In this paper we have presented work toward a self-adaptive model for evaluating and controlling QoE/QoS in networks offering multimedia services delivery. In the proposed solution, systematic service monitoring and service-oriented AC are articulated to provide consistent quality levels to multi-constrained multimedia services. The evaluation of the model's performance has demonstrated that the self-adaptive behavior inherent to online measurements combined with the proposed AC rules is effective in controlling differentiated service quality requirements. Facing the shortage of service-oriented QoS monitoring tools, a multiplatform and generic tool has been developed to provide a versatile and cost-effective QoS monitoring solution to be deployed in multiservice network environments. Future work includes developing the remaining components of the model with focus on users QoE requirements, QoE/QoS mapping and service auditing.

REFERENCES

[1] Akira Takahashi, Kazuhisa Yamagishi, and Ginga Kawaguti. Recent Activities of QoS/QoE Standardization in ITU-T SG12. *NTT Technical Review*, 6(9), September 2008.
[2] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2009–2014. White paper, June 2010.
[3] ITU-T Study Group 12. Quality of experience requirements for IPTV services. ITU-T Recommendation G.1080, December 2008.
[4] ITU-T Study Group 12. Performance monitoring points for IPTV. ITU-T Recommendation G.1081, October 2008.
[5] F. Georgatos, F. Gruber, D. Karrenberg, M. Santcroos, A. Susanj, H. Uijterwaal, and R. Wilhelm. Providing Active Measurements as a Regular Service for ISP's. *PAM'01, Amsterdam*, April 2001.
[6] http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html.
[7] Solange Rito Lima, Paulo Carvalho, and Vasco Freitas. Admission Control in Multiservice IP Networks: Architectural Issues and Trends. *IEEE Computer Communications Magazine*, 45(4):114–121, April 2007.
[8] Solange Rito Lima, Paulo Carvalho, and Vasco Freitas. A service-oriented admission control strategy for class-based ip networks. *Annales des Télécommunications*, 63(3-4):149–166, 2008.
[9] J. Babiarz, K. Chan, and F. Baker. Configuration Guidelines for Diffserv Service Classes. RFC 4594, August 2006. Informational.
[10] Neal Seitz. ITU-T QoS Standards for IP-Based Networks. *IEEE Communications Magazine*, 41(5), June 2003.