Dynamic Class of Service mapping for Quality of Experience control in future networks

by

Francesco Delli Priscoli, Laura Fogliati, Andi Palo, Antonio Pietrabissa

Department of Computer, Control and Management Engineering, University of Rome “La Sapienza”, Rome, Italy
Introduction

• The present paper derives from the work performed by the authors in the FI-WARE UE FP7 project and the PLATINO Italian PON projects, namely two major projects addressing the design of the Future Internet.

• In the authors' vision, a key Future Internet target is to allow Applications to transparently, efficiently and flexibly exploit the available network resources, aiming at meeting the personalized Users' expectations, taking into account the Present Context.

• The Users’ expectations can be expressed in terms of a properly defined Quality of Experience (QoE defined by ITU-T as: “The overall acceptability of an application or service, as perceived subjectively by the end-user“.

• Two key problems related to QoE Management are personalized QoE Evaluation and personalized QoE Control. This paper focuses on personalized QoE Control.
Future Internet Concept

Quality of Experience (QoE) Controller

Driving Parameters

Target QoE Measured QoE

Network Control
Content/Service Delivery Control
Application Control

Control decisions

Data Processing and Actuation functionalities

Users
Applications
Networks

Sensing and Data Processing functionalities

Monitored Information

Quality of Experience (QoE) Evaluator

Present Context

Context Engine

Monitored Information

Future Internet Concept
Main definition

Θ(t_s, a) : Set of feedback parameters characterizing the Present Context and relevant to the QoE experienced by the Application instance a at time t_s

QoE_{meas}(t_s, a) = g_a(Θ(t_s, a)) : Measured QoE relevant to the Application instance a at time t_s

QoE_{target}(a) : Target QoE relevant to the Application instance a

• The parameters Θ and QoE_{target}, as well as the functions g_a have to be properly selected by the QoE Evaluator and allow to personalize QoE computation and requirements.
QoE Controller objective

\[ e(t_s, a) = QoE_{meas}(t_s, a) - QoE_{target}(a) : \text{QoE Error relevant to the Application instance } a \text{ at time } t_s \]

- If the QoE Error is positive the Application is overperforming. If the QoE Error is negative the Application is underperforming.

- The scope of the QoE Controller is to avoid, as far as possible, the occurrence of underperforming applications.

- If the above-mentioned scope is not possible, the QoE Controller should try to reduce QoE Errors and to guarantee fairness.

- The QoE Controller has to dynamically select Driving Parameters which should steer the other Control procedures to take Control decisions consistent with the above-mentioned goals.
Innovations behind the proposed approach

- The fact that the Present Context (QoE Controller input) is continuously updated and the Driving Parameters (QoE Controller outputs) are dynamically selected allow fast reaction to any kind of impairment, resulting in improved performance.

- The flexibility in the selection of the parameters $\Theta$ and $QoE_{\text{target}}$, as well as of the functions $g_a$ allow QoE personalization.

- The fact that the Measured QoE is a function of the present context $\Theta$ allows network awareness driving towards fully cognition.

- The fact that both the present context and the driving parameters can be multi-network and multi-layer drives towards inter-layer optimization.

- The QoE Controller is technology independent and is decoupled both from the QoE Evaluator and from the other control functions (network control, content/service control, application control,...).
In this paper, the selected Driving Parameters are the Service Classes.

Each Service Class $k$ ($k=1,2,\ldots,K$) is characterized by pre-defined requirements (e.g. in terms of minimum guaranteed bit rate, maximum transfer delay, maximum loss bit rate, etc.) which the Network Control functionalities should satisfy by taking appropriate Control decisions.

In traditional telecommunication networks an Application instance $a$ is \textit{statically} associated to a Service Class $k(a)$ for its whole lifetime.

Conversely, in the proposed solution, the QoE Controller can \textit{dynamically} select, at each time $t_s$, the Service Class $k(t_s,a)$ which, considering its requirements and the Present Context, appears the most appropriate in order to achieve the QoE Controller goals.
Distributed architecture

• The QoE Management can be implemented by means of Agents transparently embedded in properly selected network entities: each Application instance $a$ is one-to-one associated to an Agent $a$.

• Each Agent is in charge of dynamically selecting the most appropriate Service Class for the associated Application instance on the basis of the relevant Present Context parameters.

• Key requirements for selecting the placement of the Agents are:
  – to avoid to overwhelm the network with signalling overhead;
  – to guarantee coordination among Agents.

• In the proposed solution:
  – no signalling exchange among Agents;
  – a Supervisor Agent broadcasts, at times $t_l$ (with $t_l > t_s$), a Status Signal averaging the Present Context parameters relevant to the Application instances associated to the various Service Classes $\Theta_{SA}(t_l, k), \quad k = 1, 2, \ldots K$
Each Agent $a$ is embedded a Reinforcement Learning algorithm (e.g. Q-Learning) characterized by:

- a **state set** including:
  - the *Measured QoE* evaluated by the Agent $a$ on the basis of the Present Context relevant to $a$, i.e. $QoE_{meas}(t_s, a) = g_a(\Theta(t_s, a))$
  - the *Estimate of the Measured QoE* that the Agent $a$ would experience in Service Class $k$, computed on the grounds of the Status Signal, i.e. $g_a(\Theta_{SA}(t_s, k)) \ k = 1, 2, \ldots, K$

- an **action set** including the possible Service Classes $k$ ($k=1, 2, \ldots, K$);

- a **reward function** aiming at long run maximization of
  $-[QoE_{meas}(t_s, a) - QoE_{target}(a)]^2$
• Active $\Theta_A$ and Passive $\Theta_P$ feedback parameters accounting for the ones which can be monitored with and without the user involvement respectively:
  – active parameters include the number of “clicks of blame” provided by the users;
  – passive parameters include Admitted Bit Rate, Transfer Delay, Loss Bit Rare.

• Proper relationships provide Active Measured QoE $QoE_{meas}^{ACT}(t_s,a)$ and Passive Measured QoE $QoE_{meas}^{PASS}(t_s,a)$ as a function of active and Passive feedback parameters, respectively.

• Overall Measured QoE expressed as a linear combination of passive and active measured QoE, with a coefficient $F(u)$ in the range $[0,1]$ accounting for the reliability of user $u$, i.e.:

\[
QoE_{meas}(t_s,a) = (1 - F(u))QoE_{meas}^{PASS}(t_s,a) + F(u)QoE_{meas}^{ACT}(t_s,a)
\]
Proof-of-concept scenario (2/2)

• Static and Dynamic case compared with respect to the experienced QoE Average Percentage Error, with increasing traffic conditions (from Sim1 to Sim3).

• Three Application types indicated as High, Medium and Low, making reference to the priority level of their permanently associated Service Class in the Static case (no permanent association in the dynamic case).
Conclusions

• The proposed QoE Management includes:
  – a QoE Evaluator in charge of computing, on the grounds of the Present Context (feedback information), the actual QoE perceived by the applications, as well as the expected QoE;
  – a QoE Controller which, on the basis of the QoE Evaluator outputs, has to compute proper driving parameters which should steer the other control functions towards QoE error reduction and fairness achievement.

• The QoE Controller is implemented by means of Agents including learning functionalities able to make autonomous and consistent decisions based on local and global (status signal) feedback measurements.

• The proposed cognitive approach:
  – allows to decouple QoE Evaluation, QoE Control and other control functionalities;
  – does not require a priori knowledge of the environment (model);
  – QoE requirements can be personalized;
  – signalling exchanges among agents are not required.
Future work

• Tailoring of the proposed approach to each environment:
  – mapping of the QoE Agents into the physical entities;
  – choice of the feedback parameters (Present Context);
  – proper selection of the functions for QoE evaluation (machine learning based techniques can be useful);
  – tuning of Reinforcement Learning algorithm parameters.

• Improving multi-agent coordination with minimal information exchanges (consensus based techniques are under study).
Future Internet Concept

Future Internet Core Platform

Applications

- Application Information
- User Feedbacks

Networks

- Monitoring Information

Application Interface functionalities

Cognitive Network Control functionalities

Sensing functionalities

Actuation functionalities

Present Context

Driving parameters

Control decisions

Actuation commands

Monitored Information