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

Francesco Delli Priscoli, Laura Fogliati, Andi Palo, Antonio Pietrabissa
Department of Computer, Control and Management Engineering "Antonio Ruberti", University of Rome "La Sapienza", Rome, Italy
{dellipriscoli, fogliati, palo, pietrabissa}@dis.uniroma1.it

Abstract

The present work introduces a novel approach to cope with some key limitations of the present communication networks. In particular, the need for effectively managing heterogeneous resources over heterogeneous networks while guaranteeing personalized Quality of Experience (QoE) requirements to the applications, claims for a full cognitive approach which is realized through the introduction of an appropriate *Future Internet* architecture. The paper, taking this architecture in mind, introduces the innovative concept of *dynamic* association between applications and Classes of Services, performed by means of proper Reinforcement Learning algorithms. The resulting procedure guarantees QoE personalization, requires low processing capabilities and entails limited signalling overhead.

Keywords *Quality of Experience, Quality of Service, Future Internet, Reinforcement Learning*.

1 Introduction

The FI-WARE UE FP7 project [1] and the PLATINO Italian National project [1] are two major projects which are trying to address the issues raised by the design of the Future Internet and, in particular, by the necessity to assure a personalized Quality of Experience (QoE) which represents a key Future Internet novelty.

In the authors' vision, the Future Internet overall target is to allow Applications to transparently, efficiently and flexibly exploit the available network resources, aiming at achieving a satisfaction level meeting the personalized users' QoE [2], [3]. The International Telecommunication Union (ITU-T) defines QoE as: *The overall acceptability of an application or service, as perceived subjectively by the end-user* [4]. In other words, QoE is the perception that the user has about the performance of the network when he uses an application and how this application is usable.

A large amount of research is on-going and is expected, in the near future, in the field of the identification of the personalized user expected QoE level in a given context for a given application (e.g. see [5], [6] for voice and [7], [8] for video applications, respectively), as well as of the functions for QoE computation, including passive and active monitorable feedback parameters which serve as independent variables for these functions; in particular, several works focus on studying the QoE relation with network QoS parameters [9]. By *passive* and *active parameters* we mean the ones which are independent of and dependent on the active involvement of the users in their computation, respectively [10].

This paper focuses (i) on the Future Internet *cognitive* architecture supporting QoE, (ii) on a flexible way of computing QoE on the basis of a suitable set of (passive and/or active) monitorable parameters, and (iii) on the definition of QoE Agents which, for each application, perform control functions aiming at minimizing the difference between the computed QoE and the desired QoE level.

A first key innovation of the paper is that, for a given application, the structure of the proposed QoE Agent and its way of working are flexible, since they do not impose specific requirements to the function for QoE computation and to the set of monitorable (passive and/or active) parameters.

A further fundamental innovation of the paper is that the QoE Agents, on the basis of the monitorable parameters, aim at approaching the desired QoE level of the applications by dynamically selecting the most appropriate class of service supported by the network. In this work, the selection is driven by an adaptive algorithm based on the Reinforcement Learning (RL) methodology. The proposed dynamic approach differs from traffic classification approaches found in the literature (e.g., [11] and references in [12]), based on host-level communication behaviour-based approaches, or on statistical approaches relying on data mining methodologies, since they statically determine the class of service of the application.

2 Future Internet Core Platform Architecture

This section gives a high level overview of the Future Internet Core Platform architecture, built on the work in [3], [13], [14], [15]. Figure 1 highlights some key functionalities of the Future Internet Core Platform [16]. Such functionalities can be implemented by means of distributed Agents to be transparently embedded in properly selected network nodes (e.g., Mobile Terminals, Base Stations, Backhaul Network entities, Core Network entities).

The *Sensing and Data Processing functionalities* are in charge of (i) the *monitoring* and the preliminary *filtering* of properly selected possibly heterogeneous information (e.g., including device, network performances, user profiles, network provider policies, etc.), (ii) the formal description of the above-mentioned heterogeneous information in homogeneous metadata, (iii) the proper aggregation of these metadata to form a multi-layer, multi-network *Present Context* which is a valuable input for the Network Control functionalities.

The *Cognitive Application Interface functionalities* works on the basis of (i) a pre-elaborated version of the Present Context including passive monitorable feedback parameters, (ii) the active and passive monitorable feedback parameters gathered

ISBN 978-3-8007-3602-7

from the application. The nature of the passive parameters can be related to Quality of Service (QoS) (e.g. delay, jitter, throughput, loss,…), SPD (security, privacy, dependability,…), mobility, user profiles, contents, services, or objective measurements at application level; the nature of the active parameters can be obtained by user online subjective feedback by means of suitable exposed API . The Cognitive Application Interface includes QoE Agents (detailed later) which, based on the active and passive parameters, are in charge of computing proper *driving parameters* which have to properly steer the Cognitive Network Control functionalities.

The *Cognitive Network Control functionalities* (in the following, also simply referred to as *Network Control* functionalities") consist of a set of cooperative, technology-independent algorithms and procedures which, on the basis of the Present Context provided by the Sensing functionalities and on the driving parameters provided by the Cognitive Application Interface, are in charge of taking control decisions concerning specific Network Control problems.

Finally, the *Data Processing and Actuation functionalities* are in charge of "translating" the technology-independent control decisions taken by the Network Control functionalities, in technology-dependent actuation commands which put into operation on the Networks the above-mentioned decisions.

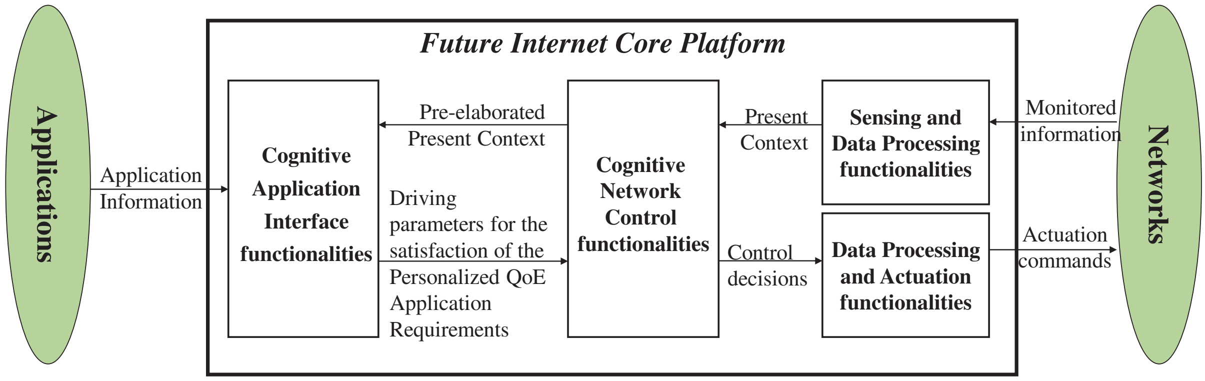

Figure 1: Future Internet Core Platform concept

The fact that both the Application Interface and the Network Control functionalities control decisions can be based on the Present Context allows *closed-loop* control, i.e. both the Application Interface and the Network Control become *cognitive*.

This paper focuses on the Agents implementing the Cognitive Application Interface; such interface is in charge of deducing the parameters which should drive the Cognitive Network Control functionalities towards the satisfaction of the Personalized QoE Application Requirements. Instead, the Cognitive Network Control functionalities are outside the scope of this paper; instances of such functionalities can be found in admission control [17], routing [18], congestion control and scheduling [19], dynamic capacity assignment [20], [21], medium access control [22].

Beside the conceptual innovations described so far, the key novelty of this paper lies in the innovative design of the Cognitive Application Interface, whose way of working is based on the application of well-known RL techniques in a multi-Agent environment consistent with the Future Internet distributed framework outlined in this section.

3 The Cognitive Application Interface

3.1 Basic definitions and samples

By **Microflow** we mean the flow of packets, relevant to a given in-progress Application instance, flowing from a given source node to a given destination node and which, in any case, is associated to a same Service Class $k \in K$ supported by the network; for instance, a given bidirectional teleconference instance taking place between two terminals A and B, is supported by 4 Microflows, i.e., two audio Microflows (from A to B and vice versa), and two video Microflows (from A to B and vice versa).

Let a denote a generic Microflow.

In a Future Internet framework each Microflow is expected to have its Personalized QoE Microflow Requirements. The Application Interface has the task of associating the Microflows with the Service Classes: as a matter of fact, at present, for scalability reasons, Network Control is performed on a per-Service Class basis. This means that for a given Microflow a, the associated Service Class $k(a)$ is the parameter which should drive the Network Control functionalities to satisfy the Personalized QoE Microflow Requirements.

At present, each Microflow a is *statically* associated (statically, means for the whole application duration) to a *Service Class* $k(a)$, properly selected in the set of K predefined Service Classes. This approach entails the following limitations:

(i) due to the very different Personalized QoE Microflow Requirements, in general, no Service Class perfectly fits the requirements of a given Microflow;

(ii) the static association between Microflows and Service Classes prevents a possible online adaptation of such association to fit the present Network status (which could be subject to sudden variations, for instance, due to traffic dynamics);

(iii) the fact that Network Control is performed on a per-Service Class basis, rather than on a *per-Microflow* basis, can entail fairness problems among Microflows associated to a same Service Class.

Let denote as t_s the time instants, periodically occurring with period T_S (i.e., $t_{s+1} = t_s + T_s$) at which Network Control takes place. In order to overcome the above-mentioned inconveniences, this paper proposes an innovative *dynamic* association between Microflows and Service Classes. So, the Cognitive Application Interface, on the basis of the Personalized QoE Microflow Requirements and of the above-mentioned feedback parameters, is in charge of selecting, at each time t_s, for each Microflow a, the most appropriate Service Class $k(t_s,a)$. The key criterion underlying the above-mentioned dynamical

ISBN 978-3-8007-3602-7

selection is to approach, as far as possible, a QoE level meeting the various Personalized QoE Microflow Requirements.

3.2 QoE computation and requirements

In the literature, a large variety of models for QoE computing and associated QoE requirements is being defined (e.g. see [5]-[8]). In this respect, the approach followed in this paper is extremely flexible since it leaves completely open the definition of the QoE computation and of the QoE requirements. According to this approach, we propose to compute the QoE and the associated requirements on the basis of:

1) **a set $\Theta_P(t_s,a)$ of selected passive feedback parameters** (monitorable without the user involvement) **and a set $\Theta_A(t_s,a)$ of selected active parameters** (directly provided by the users) which are relevant to the QoE experienced by the Microflow *a*;
2) for each Microflow *a,* **a function h_{PAS-a}** and **a function h_{ACT-a}** which compute, based on the set $\Theta_P(t_s,a)$ and $\Theta_A(t_s,a)$ respectively, the so-called *Passive* and *Active Measured QoE* experienced by the Microflow *a* at time t_s, hereafter indicated as $QoE_{meas}^{PASS}(t_s,a)$ and $QoE_{meas}^{ACT}(t_s,a)$, respectively, i.e.:

$$QoE_{meas}^{PASS}(t_s,a) = h_{PAS-a}(\Theta_P(t_s,a)) \quad QoE_{meas}^{ACT}(t_s,a) = h_{ACT-a}(\Theta_A(t_s,a)) \quad (3.1)$$

4) **a function *g***, which computes the complete *Measured QoE* based on both the *Passive* and *Active Measured QoE*s, i.e.:

$$QoE_{meas}(t_s,a) = g\left(QoE_{meas}^{PASS}(t_s,a), QoE_{meas}^{ACT}(t_s,a)\right) \quad (3.2)$$

5) **a QoE Target**, hereafter indicated as $\boldsymbol{QoE_{target}(a)}$, whose achievement entails the satisfaction of the Personalized QoE Microflow *a* Requirements.

This means that if $QoE_{meas}(t_s,a) \geq QoE_{target}(a)$ at time t_s, the Personalized QoE Microflow Requirements are satisfied.

Note that the above-mentioned approach has the key advantage of leaving completely open the definition of the complete *Measured QoE,* as well as of its Passive and Active components; so, a proper selection of $\Theta_P(t_s,a)$, $\Theta_A(t_s,a)$, h_{PAS-a}, h_{ACT-a}, *g* and $QoE_{target}(a)$ can allow to personalize the QoE computation and requirements.

Now, we can define the following ***QoE Error function***:

$$e(t_s,a) = QoE_{meas}(t_s,a) - QoE_{target}(a) \quad (3.3)$$

If the above-mentioned error is *negative*, at time t_s the Microflow *a* is not experiencing a satisfactory QoE (*underperforming Application*). If the above-mentioned error is *positive*, at time t_s the Microflow *a* is experiencing a QoE even better than expected (*overperforming Application*). Note that this last situation is desirable only if the network is idle; conversely, if the network is congested, the fact that a given Application *a* is overperforming is not, in general, desirable, since it can happen that such Application is subtracting valuable resources to underperforming applications.

In light of the above, the Cognitive Application Interface should *dynamically* determine, for each Microflow *a*, on the basis of the measured QoE and of the Personalized QoE Microflow Requirements, the most appropriate Service Class $k(t_s,a)$ to be associated to such Microflow. The *goal* of the above-mentioned selection is to avoid, as far as possible, the occurrence of underperforming Applications; in case this is not possible (e.g., due to network congestions), the dynamical selection should aim at guaranteeing fairness among Microflows from the QoE Error point of view.

It should be clear that the proposed Cognitive Application Interface can be developed independently of the Network Control functionalities, i.e., it can be used in conjunction with any type of such functionalities (either cognitive, or not) and these last can continue to operate according to their usual way of working. In other words, thanks to the proposed Cognitive Application Interface, the whole Future Internet Core Platform becomes closed-loop (i.e., cognitive) regardless of the actual way of working of the Network Control functionalities (these last can be either *open* or *closed loop*).

It is worth remarking that a proper *dynamical* selection of the most appropriate Service Class $k(t_s,a)$ allows to overcome the limitations of the *static* approach, since:

(i) the proposed approach allows complete QoE personalization thanks to the fact that $\Theta_P(t_s,a)$, $\Theta_A(t_s,a)$, h_{PAS-a}, h_{ACT-a}, *g*, and $QoE_{target}(a)$ can be freely defined;
(ii) the introduction of *cognition* in the Application Interface entails the ability to perceive possible QoE performance impairments and to consistently react;
(iii) the fact that each Microflow can have its Personalized QoE Requirements entails a per-microflow control.

3.3 Distributed architecture

As already stated in Section 2, the Agents implementing the Cognitive Application Interface have to be carefully embedded in properly selected network nodes. For the sake of simplicity, we will refer to these agents as QoE Agents. A key criterion for mapping the QoE Agents into the network nodes is to avoid overwhelming the Network with signalling overhead. So, in general, the QoE Agents relevant to various Microflows have to be embedded in different network nodes.

In turn, the facts that (i) the QoE Agents will be, in general, embedded in different network nodes, (ii) the requirement of saving signalling overhead, and (iii) the perspective of a plenty of QoE Agents simultaneously in-progress, impose the desirable requirement that no signalling exchange takes place among QoE Agents. So, these Agents should take their decisions independently one another, without exchanging information. Clearly, this entails a very complex problem of coordination among the QoE Agents considering that the decisions of a single agent impact on the network which is shared among all agents.

In order to cope with the above-mentioned coordination problem, this paper foresees the presence of a single Supervisor Agent which does not take any information from the QoE Agents, but it is in charge of broadcasting, on a semi-real-time basis, a proper Status Signal accounting for the present overall Network status, computed on the basis of the passive feedback parameters monitored by the Supervisor Agent itself. The Status Signal is sent only at times t_l periodically occurring with period T_l, with $T_l >> T_s$.

In the following, we will denote as $ss(t_l)$ the Status Signals broadcast at time t_l. Let $\Theta_{SA}(t_l,k)$ (for k=1,…,K) denote the passive feedback parameters monitored by the Supervisor Agent for each Service Class *k*. Then, the Status Signal is a string of bits:

$$ss(t_l) = [\Theta_{SA}(t_l,1), \Theta_{SA}(t_l,2),\ldots, \Theta_{SA}(t_l,K)] \quad (3.4)$$

The Supervisor Agent can be embedded either in a properly selected network node, or in an appropriate dedicated equipment (e.g., in the emerging Software Defined Networks paradigm, this could be the centralized controller [23]).

4 Modelling as a Reinforcement Learning (RL) Problem

This section details a possible implementation of the concepts described in the previous section in the perspective of embedding proper RL Algorithms in the QoE Agents. The basic issues characterizing such algorithms are the *state set*, the *reward function* and the *action set*.
As concerns the *state set*, the QoE Agent a (namely the one in charge of controlling the Microflow a), by using the function $h_{PAS\text{-}a}$ and the feedback parameters included in the Status Signal $ss(t_l)$ can compute the so-called *estimated QoE relevant to the Service Class k* (from 1 to K), hereinafter indicated as $QoE_{estim}(t_l,a,k)$, in the following way:

$$QoE_{estim}(t_l,a,k) = h_{PAS\text{-}a}(\Theta_{SA}(t_l,k)), \text{ with } k = 1,\ldots,K \qquad (4.1)$$

The parameter $QoE_{estim}(t_l,a,k)$ represents an *estimate* of the QoE that the application a will experience at times t_s next to t_l if the QoE Agent a selects the Service Class k. Note that the parameters $QoE_{estim}(t_l,a,k)$ take into account the personalized QoE of the Microflow a , since such personalization is somehow embedded in the function $h_{PAS\text{-}a}$.
To evaluate the overall Measured QoE ($QoE_{meas}(t_s,a)$), the proposed definition of the function g, is expressed by:

$$\begin{aligned} QoE_{meas}(t_s,a) = (1-F(u))QoE^{PASS}_{meas}(t_s,a) + \\ F(u)QoE^{ACT}_{meas}(t_s,a), \text{with } F(u) \in [0,1] \end{aligned}. \qquad (4.2)$$

Here, it is assumed that both the Active and the Passive Measured QoEs are normalized and assume values between 0 and 1; it follows that $QoE_{meas}(t_s,a) \in [0,1]$. The $F(u)$ parameter expresses the reliability of the user u (user trustworthiness) in producing the active parameters (and hence the reliability of the Active Measured QoE). In this respect, [10] reports a promising approach for assessing $F(u)$, even proposing a model for its dynamic evolution.
In the light of the above, a suitable state definition, at time t_s, includes $QoE_{meas}(t_s,a)$, $QoE_{estim}(t_l,a,1)$, $QoE_{estim}(t_l,a,2),\ldots$, $QoE_{estim}(t_l,a,K)$, normalized with respect to QoE_{target} and quantized. In this way, the RL algorithm of the QoE Agent a is provided with a suitable information set including (i) local information contained in the parameter $QoE_{meas}(t_s,a)$ reflecting the local situation taking place at the QoE Agent a, (ii) *global* information contained in the parameters $QoE_{estim}(t_l,a,k)$ (k=1,…,K) providing an estimate of the general network state as monitored by the Supervisor Agent, but interpreted by the QoE Agent a through its personalized function $h_{PAS\text{-}a}$.
As concerns the *reward function* $r(t_s)$, a promising choice is:

$$r(t_s) = -[QoE_{meas}(t_s,a) - QoE_{target}(a)]^2 \qquad (4.3)$$

By adopting this function, the reward is dependent on the state visited at time t_s. Whenever the measured QoE is different from the target QoE the reward is negative, and is equal to zero only when $QoE_{meas}(t_s,a) = QoE_{target}(a)$. In this way, we push the QoE Agent to drive the Service Class selection so that the Measured QoE approaches the Target QoE; in other words, we aim at minimizing both the underperforming and the overperforming situations (see also (3.3) and related comments). On the other hand, the quadratic shape of the reward function implicitly aims at assuring fairness among the Microflows.
Consistently with all the previous discussions, in each state, the *actions* that the QoE Agent a has to perform are to select, at each time t_s, the Service Class $k(t_s,a)$.

5 Simulation example

In this and in the following section, just to provide a simple proof-of-concept of the approach introduced so far, the above-mentioned derivations will be particularized to the case in which the Passive Measured QoE is computed on the basis of some Quality of Service (QoS) passive feedback parameters, and the Active Measured QoE is computed on the basis of clicks performed by the users.
As concerns the Passive Measured QoE, we have considered a very simple QoS model just including:

- the *Transfer Delay* $D(t_s,a)$ which, for the traffic relevant to the Microflow a at time t_s, is the average delay experienced by its packets from the time they go out from the source, up to the time t_s when they arrive at the destination;
- the *Admitted Bit Rate* (bps) $R_{adm}(t_s,a)$ which, for the traffic relevant to the Microflow a at time t_s, is the bit rate of traffic which can be carried from the source to the destination;
- the *Loss Bit Rate* (bps) $R_{loss}(t_s,a)$ which, for the traffic relevant to the Microflow a at time t_s, is the bit rate of traffic lost in the run from the source to the destination.

With respect to the above-mentioned parameters, we will consider the following thresholds:

- $D_{max}(a)$: Maximum *transfer delay* [s] relevant to Microflow *a;*
- $R_{adm\text{-}min}(a)$: *Minimum guaranteed bit rate* [bps] relevant to the Microflow *a;*
- $R_{loss\text{-}max}(a)$: *Maximum loss bit rate* [bps] relevant to Microflow *a;*

We will consider the following feedback parameters:

$$\Theta_P(t_s,a) = [D(t_s,a), R_{adm}(t_s,a), R_{loss}(t_s,a)] \qquad (5.1)$$

$$\Theta_{SA}(t_l,k) = [D_{SA}(t_l,k), R_{adm\text{-}SA}(t_l,k), R_{loss\text{-}SA}(t_l,k)],$$

$$k = 1,2\ldots K, \qquad (5.2)$$

where $D_{SA}(t_l,k)$, $R_{adm\text{-}SA}(t_l,k)$, $R_{loss\text{-}SA}(t_l,k)$ are the Transfer Delay, the Admitted Bit Rate and the Loss Bit Rate computed by the Supervisor Agent aggregating the measurements of all the Microflows that, at time t_l, are mapped into the Service Class k.
As concerns the personalized function $h_{PAS\text{-}a}$ allowing the computation of the Passive Measured QoE experienced by the Microflow a, the following function has been selected:

$$\begin{aligned} &QoE^{PASS}_{meas}(t_s,a) = \\ &\alpha_1 \max\left\{0, \min\left[1, 1-\frac{R_{adm-min}(a)-R_{adm}(t_s,a)}{R_{adm-min}(a)}\right]\right\} \\ &+ \alpha_2 \max\left\{0, \min\left[1, 1-\frac{D(t_s,a)-D_{max}(a)}{D_{max}(a)}\right]\right\} \\ &+ \alpha_3 \max\left\{0, \min\left[1, 1-\frac{R_{loss}(t_s,a)-R_{loss-max}(a)}{R_{loss-max}(a)}\right]\right\} \end{aligned} \qquad (5.3)$$

where α_1, α_2, α_3 are three constants in the range [0,1] to be selected according to the relative importance granted to admitted bit rate delays and losses, respectively; these constants are subject to the constraint $\alpha_1 + \alpha_2 + \alpha_3 = 1$.
As concerns the Active Measured QoE, $QoE_{meas}^{ACT}(t_s, a)$ is modelled, as in [10], with the rate of 'clicks of blame' that a (simulated) user performs to express his level of non-satisfaction.

6 Simulation results

The scope of these simulations is not to validate the results in a realistic environment, but just to show that, in the simple selected scenario, the Cognitive Application Interface arrangement described so far, in conjunction with the innovative dynamic association between Microflows and Service Classes, can yield meaningful efficiency and flexibility improvements with respect to the static case. More realistic scenario need appropriate approach (e.g., [24], [25]) to overcome the scalability problems of RL techniques.
The simulation results were performed by using the Opnet Modeler simulation tool by OPNET Technologies, a license-based SW platform that is widely used for both academic and industrial applications.
The considered network scenario is a UMTS downlink; in the Base Station, 15 Microflows are sent to 15 Receivers belonging to three types of applications, namely two classes of Video (HD TV/video streaming, video conferencing) and one FTP (File Transfer Protocol), with five instances running for each of the three types of applications. Standard Network Control procedures have been assumed, with queues in the Base Station differentiated per Class of Service and scheduled by a WFQ (Weighted Fair Queuing) mechanism. In this simulation scenario, the "*static case*", in which Classes of Service are permanently associated to the Applications, is compared to the "*dynamic case*", in which each QoE Agent dynamically selects the most appropriate Class of Service. This comparison is carried out for different levels of network congestion.
As concerns RL, a standard Q-learning algorithm was selected due to its known performance, shown in many application environments, and to its limited processing requirements.
The parameters α_1, α_2 and α_3, used to compute the $QoE_{meas}^{PASS}(t_s, a)$ from QoS parameters according to equation (5.3), are set equal to {1/3, 1/3, 1/3} for the video streams while for the FTP application they are set to {0, 1/2, 1/2} respectively. The user trustworthiness coefficient $F(u)$ has been set equal to 0.5 for any u. The time interval T_l between two Supervisor Agent measurement broadcasts is 20 s, whereas the control interval T_s of the QoE Agent is 5 s. The Q-learning algorithm parameters are set as follows: the learning rate is $1/t_s$, the discount rate γ is 0.7 and the exploration rate ε is 0.1. Simulation lengths are 700 s.
Table 1 shows some key parameters of the selected simulation scenario highlighting the main characteristics of the three Application Types, namely Video HD, FTP and Video Conf.; in the following, these three application Types, for the sake of brevity, will be referred to as *High*, *Med* and *Low*, respectively, making reference to their priority level in the static case.
Three simulation campaigns, denoted as *Sim1*, *Sim2* and *Sim3*, were performed, with increasing congestion level on the network, obtained by progressively reducing the network available capacity.

Application Type	$R_{offered}/R_{adm}/R_{adm\text{-}min}$ (Mbps)	D_{max} (s)	$R_{loss\text{-}max}$	QoE_{target}	*Service Class* [1]
Video HD	3.7/3.7/3.5	0.01	1%	0.95	High
FTP	-/-/-	0.20	0.1%	0.80	Med
Video conf.	0.65/0.65/0.5	0.10	5%	0.65	Low

Table 1: Simulation scenario – traffic parameters

A key parameter for assessing the proposed system performance is the so-called *QoE average error* defined, for a given Application Type, as the difference between the average measured QoE computed according to equation (4.2), and the QoE_{target} relevant to the considered Application Type. In the following, we will consider the even more meaningful *QoE average percentage error,* defined, for a given Application Type (High, Medium or Low), as the ratio between the average error and the QoE_{target} of the considered Application Type.
Figure 2 reports the QoE average percentage errors for the three considered Application Types (High, Med or Low), for the three considered congestion levels (Sim1, Sim2 and Sim3), for both the static and the dynamic cases.

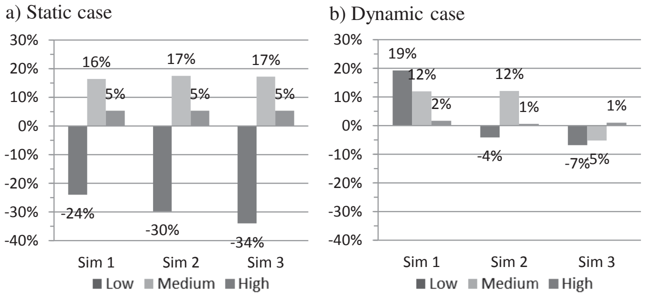

Figure 2: QoE average percentage errors

From the figure, it is evident that, in the static case, the two Application Types High and Medium are assigned a class of service that enables them to be remarkably over-performing in all the three simulation campaigns (Sim1, Sim2 and Sim3), whilst the Application Type Low is severely under-performing; this last Application Type is even more penalized as the traffic load increases (from Sim1 to Sim3).
On the contrary, in the dynamic case, the resources are utilized in a more efficient manner. In particular, in Sim1, where there is no congestion, all the Application Classes manage to over-perform. While in Sim2, where the network is slightly congested, the High and Med priority Application Classes are very close to their targets, and the Low priority Application Type meaningfully improves its performance (as a matter of fact, the QoE average percentage error is about 4%, whereas it is about 30% in the static case).
Finally, in Sim3, where the network is severely congested, only the High priority Application Class slightly over-performs, while the other two Application Types are under-performing; nevertheless, their QoE average percentage errors are much closer to zero than in the static case.
A quantitative measure of the (dramatically) improved fairness achieved by the dynamic algorithm is given by the standard deviations of the QoE average percentage errors, reported in Table 2.

[1] Classification used for the static scenarios.

ISBN 978-3-8007-3602-7

Algorithm	*Sim1*	*Sim2*	*Sim3*
Static	20.88	24.54	26.80
Dynamic	8.78	8.36	4.15

Table 2: Standard deviation of the QoE average % errors

7 Conclusions

The proposed innovative Cognitive Application Interface enables a *dynamical* Class of Service selection aimed at reducing the error between the personalized Measured QoE and the personalized target QoE levels, to let the network exploit the resources in a more efficient and flexible way. This result has been obtained by introducing learning functionalities in proper QoE Agents, which become able to make autonomous, consistent and effective decisions based on local and global (status signal) feedback measurements.

The proposed Cognitive Application Interface presents several advantages: (i) it is fully compatible with the Future Internet approach outlined in Section 2; (ii) it does not require *a priori* knowledge on the environment (model); (iii) QoE requirements can be personalized by properly selecting the functions for QoE computation and the corresponding QoE target; (iv) it is scalable since communication and cooperation/consensus/negotiation procedures among agents are not required, and the status signal information is independent on the number of agents. Future work, based on advanced regulation theory [26], is on going for improving multi-agent coordination with minimal information exchanges.

Acknowledgments

The research leading to these results was funded partly from the EU project FI-WARE (Grant Agreement n° 285248), and partly from the Italian project PLATINO (Grant Agreement n° PON01_01007). The authors wish to thank all the members of the FI-WARE and PLATINO consortia; nevertheless, the concepts and the results contained in this paper are not necessarily the ones of the consortia.

References

[1] FI-WARE (Future Internet Core Platform) EU FP7 project, contract n. 285248, www.fi-ware.eu; PLATINO (Grant Agreement n° PON01_01007).

[2] Jain, R.; Quality of experience. IEEE Multimedia, Volume: 11 , Issue: 1, 2004, pp. 96-95, DOI: 10.1109/MMUL.2004.1261114

[3] Francesco Delli Priscoli, "A Fully Cognitive Approach for Future Internet", Special Issue on "Future Network Architectures" of "Future Internet", Molecular Diversity Preservation International (MDPI), Vol. 2, January 2010, pp. 16-29.

[4] ITU-T Recommendation P.10/G.100 *New Appendix I - Definition of quality of Experience (QoE),* 2006.

[5] Gottron C., König A., Hollick M., Bergsträßer S., Hildebrandt T. and Steinmetz R. Quality of experience of voice communication in large-scale mobile ad hoc networks. In Proceedings of the 2nd IFIP conference on Wireless days (WD'09), pp. 248-253, 2009.

[6] S. Jelassi, G. Rubino, H. Melvin, H. Youssef, G. Pujolle, "Quality of Experience of VoIP Service: A Survey of Assessment Approaches and Open Issues", IEEE Communications Surveys & Tutorials, Vol. 14, Issue 2, 2012.

[7] Goudarzi P. and Nezami Ranjbar M.R. Bandwidth allocation for video transmission with differentiated quality of experience over wireless networks. Computers and Electrical Engineering 37(1), 2011.

[8] S. Singh, J.G. Andrews, G. de Veciana, "Interference Shaping for Improved Quality of Experience for Real-Time Video Streaming", IEEE Journal on Selected Areas of Communications, Vol. 30, Issue 7, Pg. 1259 – 1269, 2012.

[9] Fiedler, M.; Hossfeld, T.; Phuoc Tran-Gia; A generic quantitative relationship between quality of experience and quality of service. IEEE Network, Volume: 24 , Issue: 2, 2010, pp. 36-41, DOI: 10.1109/MNET.2010.5430142.

[10] F. Delli Priscoli, M. Iannone, A. Pietrabissa, V. Suraci, "Modelling Quality of Experience in Future Internet Networks", *Future Network & Mobile Summit 2012,* Berlin, July 2012.

[11] Karagiannis T., Papagiannaki K., and Faloutsos M. BLINC: Multilevel traffic classification in the dark. In ACM SIGCOMM, August 2005.

[12] Lee S., Kim H., Barman D., Kim C.-K., Kwon T.T., Choi Y. NeTraMark: A Network Traffic classification benchmark. In ACM SIGCOMM Computer Communication Review, 41(1), 2011.

[13] M. Castrucci, F. Delli Priscoli, A. Pietrabissa, V. Suraci, "A Cognitive Future Internet Architecture", The Future Internet Future Internet Assembly 2011, Springer Berlin/Heidelberg (DE), Lecture Notes in Computer Science, Vol. 6656, May 2011, doi: 10.1007/978-3-642-20898-0_7.

[14] M. Castrucci, M. Cecchi, F. Delli Priscoli, L. Fogliati, P. Garino, V. Suraci, "Key Concepts for the Future Internet Architecture", Future Network & Mobile Summit 2011, Warsaw, 15-17 June 2011

[15] Delli Priscoli, F., Castrucci, M., "A proposal for future internet architecture", 2010 Future Network and Mobile Summit, 16-18 June 2010, Florence,

[16] F. Delli Priscoli, V. Suraci, M. Castrucci, "Cognitive Architecture for the Internet of the Future". 6th international workshop on next generation networking middleware. October 26-30, 2009. ISBN 978-3-930736-14-0. Pag. 105-112.

[17] A. Pietrabissa, "A Reinforcement Learning Approach to Call Admission and Call Dropping Control in Links with Variable Capacity", European Journal of Control (The European Union Control Association, Lavoisier, France), Vol. 17, Issue 1, 2011, pp. 89-103, ISSN 0974-3580, DOI: 10.3166/EJC.17.89-103.

[18] C. Bruni, F. Delli Priscoli, G. Koch, A. Pietrabissa, L. Pimpinella, "Network decomposition and multi-path routing optimal control" , Transactions on Emerging Telecommunications Technologies (John Wiley & Sons, Inc., USA), Vol. 24, pp. 154-165, 2013, published on line 2 May 2012, doi: 10.1002/ett.2536.

[19] Delli Priscoli, F., Isidori, A., "A control-engineering approach to integrated congestion control and scheduling in wireless local area networks", (2005) Control Engineering Practice, 13 (5), pp. 541-558, doi: 10.1016/j.conengprac.2004.04.016.

[20] F. Delli Priscoli, A. Pietrabissa, "Design of a bandwidth-on-demand (BoD) protocol for satellite networks modeled as time-delay systems", Automatica, International Federation of Automatic Control (Elsevier, Great Britain), Vol. 40, Issue 5, pp. 729-741, May, 2004, DOI:10.1016/j.automatica.2003.12.013.

[21] R. Cusani, F. Delli Priscoli, G. Ferrari, M. Torregiani, "A Novel MAC and Scheduling Strategy to Guarantee QoS for the New-Generation Wireless LAN", Special Issue on "Mobile and Wireless Internet: Architecture and Protocols" of IEEE Wireless Communications (IEEE Personal Communications), IEEE's Computer and Vehicular Technology Societies (U.S.A.), Issue 3, June 2002, pp. 46-56.

[22] Francesco Delli Priscoli, "A Control Based Solution for Integrated Dynamic Capacity Assignment, Congestion Control and Scheduling in Wireless Networks", European Journal of Control, European Union Control Association (EUCA), Issue n.2/2010, pp. 169-184.

[23] A. Palo, L. Zuccaro, A. Simeoni, V. Suraci, L. Musto, P. Garino, "A Common Open Interface to Programmatically Control and Supervise Open Networks in the Future Internet", Future Networks & Mobile Summit 2013, Lisbon, July 2013.

[24] A. Pietrabissa, "Admission Control in UMTS Networks based on Approximate Dynamic Programming", European Journal of Control (The European Union Control Association, Lavoisier, France), Vol. 14, N. 1, pp. 62-75 , January 2008, DOI:10.3166/ejc.14.62-75.

[25] A. Pietrabissa, "An Alternative LP Formulation of the Admission Control Problem in Multi-Class Networks", IEEE Transaction on Automatic Control, (IEEE Control System Society, USA), Vol. 53, N. 3, pp. 839-845, April 2008, DOI: 10.1109/TAC.2008.919516.

[26] F. Delli Priscoli, A. Isidori, L. Marconi, "A Dissipativity-based Approach to Output Regulation of Non-Minimum-Phase Systems", System and Control Letters, Elsevier Science Pub., Vol. 58, 2009, pp. 584-591.