Approaches for Future Internet architecture design and Quality of Experience (QoE) Control

S. BATTILOTTI[1], F. DELLI PRISCOLI[1], C. GORI GIORGI[1], A. PIETRABISSA[1], S. MONACO[1], M. PANFILI[1], S. CANALE[1], V. SURACI[1,2]

[1]University of Rome "Sapienza", Via Ariosto 25, Rome, 00185, ITALY,
email: {battilotti,dellipriscoli,gorigiorgi,pietrabissa,Monaco,panfili,canale}@dis.uniroma1.it

[2]Engineering Department, Università degli studi e-Campus, Via Isimbardi 10, Novedrate, 22060, ITALY, email: vincenzo.suraci@uniecampus.it

Abstract: - Researching a Future Internet capable of overcoming the current Internet limitations is a strategic investment. In this respect, this paper presents some concepts that can contribute to provide some guidelines to overcome the above-mentioned limitations. In the authors' vision, a key Future Internet target is to allow applications to transparently, efficiently and flexibly exploit the available network resources with the aim to match the users' expectations. Such expectations could be expressed in terms of a properly defined Quality of Experience (QoE). In this respect, this paper provides some approaches for coping with the QoE provision problem.

Key-Words: Quality of Experience, Quality of Service, Future Internet, Reinforcement Learning.

1 Introduction

Future Internet design is one of the current priorities established by the UE. The FI-WARE FP7 project [1] and his FI-CORE follow-on is currently trying to address the issues raised by such design. At Italian level, the Future Internet design is addressed in the framework of the PON PLATINO project [2].

This paper is just based on the work performed by the authors in the framework of the PLATINO project.

In the authors' vision [28], a key Future Internet target is to allow applications to transparently, efficiently and flexibly exploit the available resources, aiming at achieving a satisfaction level meeting the personalized Users' needs and expectations [3], [4], [5]. Such expectations could be expressed in terms of a properly defined Quality of Experience (QoE), which, in the most general case, could be regarded as a personalized function of a plenty of parameters of heterogeneous nature and spanning all layers of the protocol stack (e.g. such parameters can be related to Quality of Service (QoS), security, mobility, contents, services, device characteristics…). In this respect, a large amount of research is on-going in the field of the identification of the personalized user expected QoE level in a given context for a given application (e.g. see [6], [7] for voice and [8], [9] for video applications, respectively), as well as of the functions for QoE computation, including the 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 [10].

In order to achieve the above-mentioned target in an efficient and flexible way, the Future Internet should overcome the following main limitations:

a) A first limitation is inherent to the traditional layering architecture which forces to keeping algorithms and procedures, lying at different layers, independent one another; in addition, even in the framework of a given layer, algorithms and procedures dealing with different tasks are often designed independently one another. These issues greatly simplify the overall design of the telecommunication networks and greatly reduce processing capabilities, since the overall of network control problem is decoupled in a certain number of much simpler sub-problems. Nevertheless, an obvious limitation of this approach derives from the fact that algorithms and procedures are poorly coordinated, impairing the efficiency of the overall network control. The issues above claim for a stronger coordination between algorithms and procedures dealing with different tasks.
b) A second limitation derives from the fact that, at present, most of the algorithms and procedures embedded in the telecommunication networks

are *open-loop*, i.e. they are based on off-line "reasonable" estimation of network variables (e.g. offered traffic), rather than on real-time measurements of such variables. This limitation is becoming harder and harder, since the telecommunication network behaviours, due to the large variety of supported services and the rapid evolution of the service characteristics, are becoming more and more unpredictable. This claims for an evolution towards closed-loop algorithms and procedures which are able to properly exploit appropriate real-time network measurements. In this respect, the current technology developments which assure cheap and powerful sensing capabilities favour this kind of evolution.

c) A third limitation derives from the large variety of existing heterogeneous underlying networks which have been developed according to different heterogeneous technologies and hence embedding technology-dependent algorithms and procedures, as well as from the large variety of heterogeneous users. In this respect, the requirement of virtualising these networks and users so that they can be dealt with in an homogeneous way by the applications, claims for the design of a technology-independent, virtualized framework.

d) A fourth limitation derives from the inability to satisfy *personalized* QoE requirements. As a matter of fact, most of the current approaches are based on the presence of a limited number of Class of Services. Each Class of Service provides given performance guarantees (e.g. in terms of QoS): then, each connection is statically mapped on the most appropriate Class of Service. Nevertheless, the requirement of satisfying a larger and larger number of new applications, as well as to meet, even for the same application, personalized user expectations claims for the overcoming the Class of Service concept and for the handling of resource assignment in a more dynamic and personalized way.

This paper outlines how Future Internet can overcome these limitations.

In this respect, first of all (Section II) this paper highlights the general high level Future Internet architecture introduced in [4], [14], [15], [16], [28], [48] and formalized within the FI-WARE project [17], showing how QoE Management is embedded in such architecture and how it contributes to achieve some key Future Internet innovations. In particular, as further detailed in Section II, the proposed QoE Management contributes to achieve the following important Future Internet goals corresponding to the limitations listed above: (a) (c) the overcoming of the traditional layering architecture and the full interoperation among heterogeneous networks, with the consequent i*nter-layer* and *inter-network* algorithms and procedures optimization; (b) the achievement of *fully cognitive* solutions with all algorithms and procedures based on *feedback parameters* coming from the monitoring of the network performance and/or by direct/indirect user feedbacks and/or by specific application requirements; (d) the personalization of the requirements to be satisfied which depend on the given user who is using a given application in a given context.

Afterwards (Sections III, IV), this paper focuses on the QoE Controller which plays a fundamental role for overcoming the above-mentioned limitations and hence achieving a satisfaction level meeting the personalized Users' expectations, while efficiently and flexibly exploiting the network resources.

As detailed in the following, the QoE Controller assesses the so-called *Driving Parameters*, namely the parameters that should drive the various Control functionalities (Network Control, Content/Service Delivery Control and Application Control) towards the minimization, for each in progress application, of the difference between the QoE expected by the user (namely, the so-called *Target QoE*) and the current QoE perceived by the user (namely, the so-called *Perceived QoE*).

It is worth remarking that this paper does not intend to provide solutions to the plenty of problems set by Future Internet design, but just to provide some hints both from an architectural perspective (Section II), and from the specific QoE point of view (Sections III and IV), thus providing some preliminary guidelines to the huge work which is expected in the next decade in this area.

2 High level future internet architecture

This section gives an overview of the proposed Future Internet concept which is sketched in Fig. 1.

A first basic concept highlighted in the figure is the decoupling between *Specific* (*Technology Dependent*) functionalities and *Generic* (*Technology Independent*) functionalities.

The Specific functionalities are the ones included in the thick box in the right part of the figure, while the Generic functionalities are included in the thick box in the left part of the figure. The concept underlying the Generic functionalities is that, following a

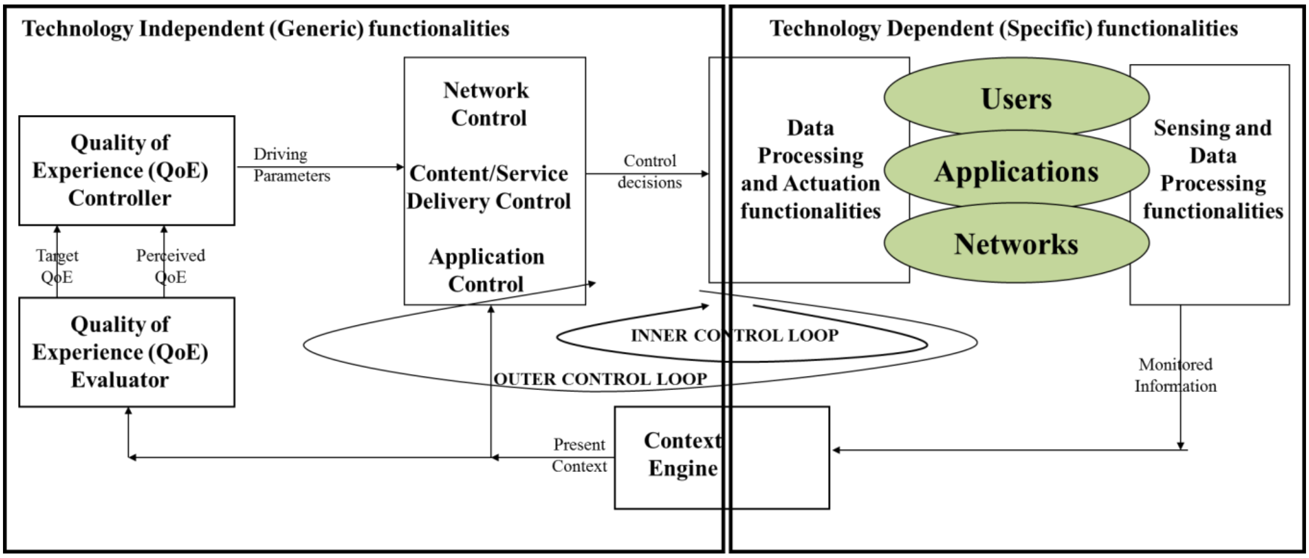

Figure 1: Proposed Future Internet approach

possible ad hoc configuration, they can be reused in conjunction with any Specific functionality. In the Future Internet terminology, the Generic (Technology Independent) functionalities are referred to as Generic Enablers [1] just to underline their general-purpose nature. In this respect, in order to overcome the limitation (c), i.e. to favour a simple interoperability among heterogeneous Specific Networks, Future Internet aims at keeping, as far as possible, most functionalities independent of the specific technology.

2.1 Specific (Technology Dependent) functionalities

The *Specific Networks* shown in Fig. 1 represent the present and the near future *Wireless/Wired Networks* characterized by *specific* technologies, as well as their *specific* control and management procedures.

A plenty of *Specific Applications* are running over those Networks offering a broader and broader range of specific services to a plethora of *Specific Users* characterized by specific profiles and accessing the network through specific devices.

A set of *Specific Sensing and Data Processing* functionalities are in charge of the real-time monitoring of the Specific Networks/Applications/Users producing the so-called *Monitored Information*. In particular, these functionalities are in charge of monitoring:

- the Specific Networks by measuring and pre-processing Network-Specific Performance Levels expressed according to *Network Specific Metrics* related to *QoS* (e.g. parameters relevant to Delay, Loss, Throughput…), *SPD* (i.e.. parameters relevant to Security, Privacy, Dependability) and *Mobility* (i.e. parameters relevant to roaming and handover). These performance levels can be exploited by Specific network control and management procedures (for the sake of clarity, not shown in the figure), as well as by the Generic control procedures (detailed in the following);
- the Applications by measuring and pre-processing their characterizing parameters (e.g. transaction frequency, transaction duration, transaction specific features…);
- the Users by measuring and pre-processing the parameters characterizing their environment (e.g. the user location, user device characteristics, etc.), as well as the User reactions while using each Application (e.g. through appreciation or blame clicks).

The Monitored Information produced by the *Specific Sensing and Data Processing* functionalities are exploited either by Specific network control and management procedures (not shown in Figure 1), or by the Generic control procedures detailed below.

Likewise, a set of *Specific Data Processing and Actuation* functionalities are in charge to put into effect on the Networks/Applications/Users the *Control Decision* taken either by the Specific network control and management procedures, or by the Generic control procedures detailed below; since these last are Technology Independent Control Decision, the Specific Data Processing and Actuation functional block is in Figure 1partially included among the Generic functionalities.

2.2 Generic (Technology Independent) functionalities: the Inner Control Loop

The control based vision of the Future Internet concept is shown in Fig. 1 highlighting the two main closed control loops involving the Generic Enablers, namely an *Inner Control Loop* and an *Outer Control Loop*. It should be clear that such closed control loops allow to overcome the limitation (b) presented in the introduction.

In particular, the Inner Control Loop (for the Outer Control Loop see the next section) consists of (i) the Users/Networks/Applications which can be regarded as the control-loop plant, (ii) the Sensing and Data Processing functionalities which, together with the Context Engine functionalities can be regarded as the control-loop sensors producing the *Present Context* (this last can be regarded as the feedback variables), (iii) the Network/Content/Service/ Application Control functionalities (hereinafter, for the sake of brevity, simply referred to as *Control functionalities*) which can be regarded as the Inner Control Loop controllers, (iv) the Data Processing and Actuation functionalities which can be regarded as the control-loop actuators.

The *Context Engine* receives the Monitored Information, i.e. heterogeneous multi-layer, multi-network information (these last being Technology Dependent Information, this is why in Fig. 1 this functional block is partially included among the Specific functionalities). Then, the Context Engine is in charge of (i) the formal description of the heterogeneous Monitored Information in homogeneous metadata (e.g. according to proper semantic language), (ii) the further processing of these metadata and their proper aggregation to form a multi-layer, multi-network technology-independent Present Context. This last should somehow "summarize" in the most compact, but still meaningful way, the present network, user and application status, thus being a key valuable feedback input for all control functionalities. It is worth stressing that such Present Context should have an highly dynamic nature.

The *Control functionalities* consist of a set of modular, technology-independent, interoperating Generic Enablers which operate, on the basis of (i) the Present Context which includes the feedback parameters and (ii) the Driving Parameters, which, as explained in the next Section, include the reference parameters which the Control functionalities should track. On the basis of the above-mentioned inputs, the *Control functionalities* have to generate *Control Decisions* aiming at (i) controlling the Networks (and, in particular, the utilization of their resources), (ii) providing the most appropriate data/services/contents to the Users, (iii) allowing to properly drive and configure the Applications.

For instance, on the basis of Driving Parameters specifying the target Quality of Service and the target content mix which should satisfy a given user, the Control functionalities decide, taking into account the Present Context, how/where to retrieve the desired contents, how/where to aggregate/enrich them, which underlying network is the most appropriate for content delivery, which resources have to be reserved on the selected network in order to delivery the contents with the desired Quality of Service (QoS), etc.

Note that, thanks to the aggregated Present Context provided by the Context Engine, the Control functionalities have a technology-neutral, multi-layer, multi-network vision of the surrounding Users, Networks and Applications, whilst, thanks to the Driving Parameters provided by the QoE Controller, the Control functionalities have reference target values they should aim to reach. In particular, the cognitive nature of the metadata which form the Present Context, coupled with a proper design of the Generic Enablers implementing the Control functionalities (e.g. multi-objective advanced control and optimization algorithms could be adopted), can lead to cross-layer and cross-network optimization, thus overcoming the limitations (a), (b), (c); moreover, as explained in the next sections, a proper handling of the QoE Management and of the related Driving Parameters can lead to overcome the limitation (d).

This paper focuses on the Outer Control Loop controller, namely the QoE Controller, whilst the Control functionalities of the Inner Control Loop are outside the scope of this paper. Instances of such functionalities can be found in admission control [17], [25], [26], [41] routing [19], [34], [36], congestion control and scheduling [20], resource discovery [44] dynamic capacity assignment [21], [22], medium access control [23], load balancing [35], [42] security [43], [50-51] and energy ([37-40], [46]).

2.3 Generic (Technology Independent) functionalities: the Outer Control Loop

The Outer Control Loop, in addition to the functional blocks already described with respect to the Inner Control Loop, also includes (i) a so-called *QoE Evaluator* which can be regarded as a further sensor functionality, (ii) the so-called *QoE Controller* which can be regarded as the Outer control-loop controller.

The QoE Evaluator is in charge of the following tasks:

- storing the Present Context in a *Knowledge Database* together with the corresponding Control Decisions taken by the Control functionalities. The *big data* stored in such Knowledge Database should be analyzed, by means of appropriate machine learning techniques to infer a number of important information (e.g. the identification of the User Profiles);
- identifying (not in real-time) a set of *N* different *User Profiles*, by analyzing (e.g. through machine learning techniques) the big data stored in the Knowledge Database: each User Profile clusters the users characterized by a similar behaviour (and interested to similar performance aspects) while using given applications;
- identifying (off-line) a set of *M* personalized (according to the application typology and the user profile) *QoE Evaluation Functions* able to assess, in real-time, for each user using a given application typology the so-called *Perceived QoE*, namely its currently experienced QoE. The very critical identification of these functions can be performed according to the following approach (i) identifying a set of *P QoE Evaluation Function Structures*, typically associated to the various application typologies, which can be deduced according to both empirical results and theoretical results taken from literature (e.g., [47] in case of streaming VOIP); each of these Function Structures should be function of both a suitable subset of the Present Context parameters, and of a proper set of *User Profile Parameters*, (ii) identifying (not in real-time), for each of the *N* User Profiles, the associated set of User Profile Parameters (this can be done, through proper machine learning algorithms, analyzing the relevant records of the Knowledge Database). By so-doing, we obtain *M=PN QoE Evaluation Functions* which are dynamic functions of the Present Context, this last being a dynamic information which can be personalized even at a single user level;
- computing (off-line) *M Target QoE*, namely the target QoE performance levels associated to the M QoE Evaluation Functions. The Target QoE should be identified by using proper machine learning algorithms able to analyze the data stored in the Knowledge Database. These targets, as well as the User Profile Parameters can be continuously updated (not in real-time) in order to exploit the always increasing and updated information stored in the Knowledge Database.

In light of the above, the output of the QoE Evaluator (i.e. the input of the QoE Controller) is, for each user running a given application, its current Perceived QoE and the associated Target QoE.

The QoE Controller (namely the controller of the Outer Control Loop) has to deduce, in real-time, the Driving Parameters which the Control functionalities (namely the controllers of the Inner Control Loop) has to track. The goal of the QoE Controller is the satisfaction of the personalized user QoE requirements, namely the minimization, for each in progress application of its *QoE Error*, defined as the difference between the Target QoE and the Perceived QoE of the user running the application in question.

To reach the above-mentioned goal, the QoE Controller should know – or, at least, estimate – the correlation between its decisions (the selected driving parameter) and the Perceived QoE in a given Present Context. In this respect, no model of the Inner Control Loop can be assumed, since it depends on too many unpredictable factors (e.g. traffic characteristics, network topologies, control functionalities, and so on). The decision strategy must therefore be learned on-line by trial and errors. In this respect, in the next section Reinforcement Learning (RL) is proposed as the key technology to enable an organized on-line exploration of the possible decision strategies, named *policies*, and the exploitation of the best policy to be enforced.

The QoE Controller can be implemented by means of Agents (referred to as *QoE Agents*) to be carefully embedded in properly selected network nodes (e.g., Base Stations and Mobile Terminals in a wireless environment).

3 Approaches for qoe controller design

In this paper we present two alternative algorithms to implement the QoE Controller. The first one, referred to as *single-agent learning*, proposes that the decisions (i.e., the value of the Driving Parameters) are taken by each Agent on the basis of its local knowledge of the Present Context and of the so-called *Status Signal*, which represents in a concise way the overall Network status, broadcast by a single centralized entity, named Supervisor Agent. In the second algorithm, referred to as *multi-agent learning*, the Agents communicate their QoE Error to the Supervisor Agent, which computes and

broadcasts the decisions; the relevant problem can be modelled as a Multiagent System [32].
In both algorithms the learning approach consists in a model-free adaptive feedback approach: the effect of the decisions are observed as a variation of the QoE Error, and the decisions are taken based on past-observations. Reinforcement Learning (RL) is a promising approach to solve both single and multiple agents problem, even though other advanced approaches are possible [27]. Both approaches entail the presence of a centralized entity, which sends control signalling to the Agents. This approach is well-matched to the current trends in managing communication network, as with the Software Defined Network [24].
Concerning the Driving Parameters, their nature depends on the considered application typology and user profile. For instance, the Driving Parameters can include, among others, Quality of Service (QoS) reference values (e.g., these QoS reference values could concern the tolerated transfer delay range, the minimum throughput to be guaranteed, the tolerated packet loss range, the tolerated dropping frequency range, etc.), Security reference values (e.g., the expected encryption level, the expected security level of the routing path computed by introducing appropriate metrics, etc.), or Application-specific reference values (e.g. the expected video resolution, the expected audio encoding, etc.), or Load Balancing reference values (the expected distribution of the offered traffic among the heterogeneous wireless access networks simultaneously covering a given user).
In the following of this paper, we refer to the case in which the Driving Parameters are QoS reference values: in this case the QoE Controller has to dynamically decide, for each running application, the most appropriate QoS reference values which, thanks to the Control functionalities performed in the Inner Control Loop, should drive the Perceived QoE as close as possible to the Target QoE. Since the control action has a large number of degree of freedom, the solution space exploration may take a large amount of time: so, the QoE Controller task may be complex. A simpler control task arises if QoS management of the underlying network is organized in Classes of Service (CoS). In this case, the role of the QoE Controller is to dynamically select, in real-time, the most appropriate CoS for the on-going applications (i.e. the Driving Parameters are the CoSs associated to the running applications) aiming at reducing the QoE Error.

3.1 Single Agent Reinforcement Learning

The problem is described by a Markov Decision Process, a tuple $\{X, A, pr, r\}$, where r where is the finite state space, A is the finite set of agent actions, pr is the transition probability function, r is the one-step reward function. The state $x \in X$, that describes the environment, can be altered by the agent action $a \in A$. The environment changes state according to the state transition probabilities given by $pr(x, a, x')$. The reward evaluates the immediate effect of action a. The behaviour of the agent is described by its policy π, which specifies how the agent chooses its actions given the state, it may be either stochastic, $\pi: X \times A \rightarrow [0,1]$, or deterministic, $\pi: X \times A$.
We consider a common reinforcement learning technique, known as Q-Learning [29], [30], that works by learning the action-value function. The action-value function $Q^{\pi}(x, a)$ is the expected return starting from x, taking action a, and thereafter following policy π; it satisfies the Bellman equation:

$$Q^{\pi}(x,a) = \sum_{x' \in X} pr(x,a,x') \left[r(x,a,x') + \gamma \max_{a' \in A} Q^{\pi}(x',a') \right] \quad (1)$$

where the discount factor $\gamma \in [0,1)$ weights immediate rewards versus delayed rewards.
Let $Q^{*}(x, a)$ be the optimal action-value function, defined as:

$$Q^{*}(x,a) = \max_{\pi} Q^{\pi}(x',a'), \forall x \in X, a \in A(x) \quad (2)$$

Then, the agent, computing $Q^{*}(x, a)$, can maximize its long-term performance, while only receiving feedback about its immediate, one-step performance. The greedy policy is deterministic and picks for every state the action with the highest Q-value:

$$\pi(x) = \arg \max_{a' \in \mathrm{A(x)}} Q(x,a') \quad (3)$$

The Q-learning approach derives the policy on-line by estimating the (action, state)-values with the following update rules:

$$Q(x,a) \leftarrow \left(1 - \alpha(x)\right) Q(x,a) + \alpha(x) \left[r(x,a,x') + \gamma \max_{a' \in \mathrm{A}} Q(x',a') \right] \quad (4)$$

where the learning rate $\alpha(x) \in [0,1]$ determines the convergence speed and accuracy.

3.2 Multiagent Reinforcement Learning

As in [49], the generalization of the Markov Decision Process to the multiagent case is a stochastic game (SG) described by a tuple $\{X, A_1, \dots, A_N, pr, r_1, \dots, r_N\}$ where N is the number of agents, X is the discrete set of environment states, A_n is the discrete sets of actions available to the agent $n, n = 1, \dots, N$, yielding the joint action set $A = A_1 \times \dots \times A_N$, $pr: X \times A \times X \to [0,1]$ is the state transition probability function, and $r_n: X \times A \times X \to \mathbb{R}$ is the reward functions of the agent $n, n = 1, \dots, N$. The state transitions and the reward depend on the joint action of all the agents, $\mathbf{a} = [a_1^{\mathrm{T}}, \dots, a_N^{\mathrm{T}}], \mathbf{a} \in A, a_n \in A_n, n = 1, .., N$. The policies $\pi_n: X \times A_n \to [0,1]$ form together the joint policy $\boldsymbol{\pi}$. Clearly, the Q-function of each agent n (Q_n^{π}), depends on the joint action and is conditioned on the joint policy:

$$Q_n(x, a_1, \dots, a_N) = \sum_{x' \in X} pr(x, a_1, \dots, a_N, x')[r_n(x, a_1, \dots, a_N, x') + \gamma Q_n(x', \pi_1, \dots \pi_N)], n = 1, \dots N \quad (5)$$

where $Q_n(x', \pi_1, \dots \pi_N)$ is a weighted sum of $Q_n(x, a_1, \dots, a_N)$.

Considering the single agents Q-learning approach (4), it is possible to define an analogue approach for Multiagent RL as follow:

$$Q_n(x, \mathbf{a}) \leftarrow (1 - \alpha(x))Q_n(x, \mathbf{a}) + \alpha(x)\left[r_n(x, \mathbf{a}, x') + \gamma\, \mathrm{eval_n}\left(\boldsymbol{\pi}(x')Q(x', \pi_n(x'))\right)\right], n = 1, \dots N \quad (6)$$

$$\boldsymbol{\pi}(x) = \mathrm{solve}_{\pi}(Q_1(x, \mathbf{a}), \dots, Q_N(x, \mathbf{a})) \quad (7)$$

where solve_{π} is a selection mechanism mapping from one stage games into joint distributions and $\mathrm{eval_n}$ gives the expected return of agent given this joint distribution.

Littman in [31] presents a convergent algorithm, denoted friend-or-foe Q-learning (FFQ), that, in fully cooperative SG (e.g. $r_1 = \dots = r_N$) or fully competitive SG (i.e. $r_1 = -r_2$), converges to the value Nash-Q [32]. Furthermore, in fully cooperative SG, if a centralized controller were available, the task would reduce to a Markov decision process (the action space would be the joint action space of the SG) and the goal could be achieved by learning the optimal joint-action values with simple Q-learning:

$$Q(x, \mathbf{a}) \leftarrow (1 - \alpha(x))Q(x, \mathbf{a}) + \alpha(x)\left[r(x, \mathbf{a}, x') + \gamma \max_{\mathbf{a}' \in \mathrm{A}} Q(x', \mathbf{a}')\right] \quad (8)$$

3.3 Problem statement

A generic network with the following features is considered:

1) available link capacity, denoted with B_{link};
2) M Application types, each one characterized by an average transmission bit rate b_m, $m = 1, \dots, M$;
3) N end-nodes/agents, each one supporting one particular application and characterized by personalized Target QoE level denoted $TQoE_n$, $n = 1, \dots, N$.

It is assumed that the network supports classes of service. At each time step t, each agent n selects the most appropriate service class to be associated with the application supported by the node in question. We define $a_n(t)$, $n = 1, \dots, N$, the control action of node n at time step t. Let $\mathbf{a}(t)$ be the vector of control action of all nodes, i.e.:

$$\mathbf{a}(t) = (a1(t), \dots, a_N(t)), \text{ where } a_n(t) \in \{1, \dots C\} \quad (9)$$

The control objective is to minimize the error between the measured Perceived QoE, denoted $PQoE_n$, and the QoE target, for each node n.

4 QoE controller algorithms

Two multi agent RL approaches are proposed to solve the problem defined in the previous section. In both approaches, a soft method can be considered in order to address the exploration problem. In particular, ε-greedy policy is a soft method that consists in the selection of a random action with a small probability; in details, it selects: i) with probability $1 - \varepsilon$, the greedy action (7), and ii) with probability ε, a random action $a \in A$, where the parameter $\varepsilon \in [0,1]$ weights the exploration of the state-space versus the exploitation of the current estimates of the (action, state)-values.

4.1 Single Agent Reinforcement Learning Approach

In the single-learning algorithm, at each decision period each Agent tries to minimize its QoE error by deciding its CoS for the next time interval, based on the local feedback on the available transmission rate, and on the Status Signal, which communicates the number of Agents which currently opted for each CoS. The decision is based on the estimate of the expected QoE error which may be achieved by switching to a given CoS. In this approach the single agent Q-learning is directly applied to the multi agent case, thus the joint actions are not consider. In order to model all information to solve the problem, we define the following Markov decision processes $\{X, A, pr, r_n\}$, for each agent n:

1) The space state X describes the environment; considering that, the state represents the vector of active nodes enjoying the service m, $m = 1, \dots, M$, using the class of service c, $c = 1, \dots, C$, at time t: $x(t) = (n_{11}(t), \dots, n_{1C}(t), \dots, n_{M1}(t), \dots, n_{MC}(t))$, where $n_{cm} = 0,1, \dots$; $c = 1, \dots, C$; $m = 1, \dots, M$. Thus the finite state space is defined as $X = \{x = (n_{mc}), c = 1, \dots, C; m = 1, \dots, M\}$
2) The action set represents, for each agent, the class selected for the transmission: $A_n = A = \{1, \dots, C\}, n = 1, \dots, N$
3) pr is the transition probability function;
4) For each agent n the cost $r_n(x, a, x')$ is defined by the error between the Perceived QoE $PQoE_n(x, a, x')$, and the Target QoE of agent n, $TQoE_n$: $r_n(x, a, x') = |\, PQoE_n(x, a, x') - TQoE_n|$, $n = 1, \dots, N$

In this case each agent solves an independent Q-learning algorithm, thus from (4):

$$Q_n(x,a) \leftarrow (1 - \alpha(x))Q_n(x,a) + \alpha(x)\left[r_n(x,a,x') + \gamma \max_{a' \in A} Q_n(x',a')\right] \quad (10)$$

4.2 Multiagent Reinforcement Learning: Friend Q-Learning

In the multi-learning algorithm, the Supervisor Agent tries to minimize the average square QoE Error of the Agents by deciding their CoS for the next time interval. The decision is based on the estimate of the expected average square QoE error which is achieved by switching to a given CoS; the estimates are updated based on the QoE error measures sent by the Agents.

In this approach a static game is considered, it means a SG with $X = \emptyset$, in which the reward depends only on the joint actions. In particular we consider the following static game $\{A_1, \dots A_N, r_1, \dots, r_N\}$ where N is the number of agents, $A_1 = \dots = A_N = \{1, \dots C\}$ are the discrete sets of actions available to the agents, yielding the joint action set $A = A_1 \times \dots \times A_N = \{\mathbf{a} = [a_1^{\mathrm{T}}, \dots, a_N^{\mathrm{T}}], \mathbf{a} \in A, a_n \in A_n, a = 1, \dots N\} = \{1, \dots, C\}^N$ and $r_n : A \to \mathbb{R}, n = 1, \dots N$ are the cost functions of the agents. For each agent n the cost $r_n(\mathbf{a})$ is defined by the error between the Perceived QoE $PQoE_n(\mathbf{a})$, that the agent n achieves when the joint action $\mathbf{a}$ is taken, and the Target QoE of agent n, $TQoE_n$:

$$r_n(\mathbf{a}) = |\, PQoE_n(\mathbf{a}) - TQoE_n|, n = 1, \dots, N \quad (11)$$

Thus, the MARL approach could be described by the following equation derived by eq. (6) and eq. (7):

$$Q_n(\mathbf{a}) \leftarrow (1 - \alpha)Q_n(\mathbf{a}) + \alpha\left[r_n(\mathbf{a}) + \gamma\, \mathrm{eval_n}(\boldsymbol{\pi} Q(\pi_n))\right] \quad (12)$$

$$\boldsymbol{\pi} = \mathrm{solve}_\pi(Q_1(\mathbf{a}), \dots, Q_N(\mathbf{a})) \quad (13)$$

where solve_π returns a particular type of equilibrium and $\mathrm{eval_n}$ gives the expected return of agent n given this equilibrium.

Friend Q-learning approach converges if the SG has at least one coordination equilibrium. The coordination equilibrium is a particular Nash equilibrium, in which all players achieve their highest possible value:

$$r_n(\pi_1, \dots \pi_N) = \max_{a_1 \in A_1, \dots, a_N \in A_N} r_n(a_1, \dots, a_N), n = 1, \dots, N \quad (14)$$

If the SG is fully cooperative (e.g. $r_1 = \dots = r_N$) then there is at least one coordination equilibrium. Thus, in order to guarantee the convergence to coordination equilibrium, it is necessary to modify the SG definition provided in the previous section such that $r_1 = \dots = r_N$. One possible way to modify the static game $\{A_1, \dots A_N, r_1, \dots, r_N\}$ is to consider a new cost function:

$$r_1' = \dots = r_N' = f(r_1, \dots, r_N); \quad (15)$$

where r_n is defined in eq. (11) and an example of function f could be the Euclidean norm ($f = \|\cdot\|_2$).

Considering the static fully cooperative game $\{A_1, \dots A_N, r_1', \dots, r_N'\}$, the Friend Q-learning update rule is:

$$Q_n(\mathbf{a}) \leftarrow (1 - \alpha)Q_n(\mathbf{a}) + \alpha\left[r_n'(\mathbf{a}) + \gamma \max_{\mathbf{a}' \in A} Q_n(\mathbf{a}')\right] \quad (16)$$

Note that $Q_1 = \dots = Q_N$, thus considering a centralized entity, the original SG problem can be reduced to a Markov decision process:

$$Q(\mathbf{a}) \leftarrow (1 - \alpha)Q(\mathbf{a}) + \alpha\left[r'^{(\mathbf{a})} + \gamma \max_{\mathbf{a}' \in A} Q(\mathbf{a}')\right] \quad (17)$$

5 Conclusion

The paper presents promising architectural approaches for designing a Future Internet framework which allows to overcome the limitations (a-d) listed in the introduction. In particular, the paper defines a modular, cognitive, access-agnostic architecture which decouples the QoE Management problem from the other Control functionalities. Such a decoupling is realized through two nested *closed* control loops: namely an Inner Control Loop including the Control functionalities and an Outer Control Loop which should drive the Inner Control Loop with the aim to satisfy personalized QoE requirements (i.e. one of the most challenging goal of the Future Internet).

In turn, the QoE Management functionalities are decoupled in QoE Evaluation and QoE Controller which can be designed independently of one another. The strength of the proposed approach derives from its flexibility and from its self-adaptation ability. The QoE Evaluator does not know the Target QoE and the Perceived QoE a priori, but learn them while operating. Likewise, the QoE Control learns the most effective Driving Parameters to minimize the QoE Error between the Target and the Perceived QoE. As concerns scalability of the proposed approach, both QoE Evaluation and QoE Control are organized in such a way that all heavy (from a computing point of view) learning tasks are performed either off-line, or not in real-time.

In particular, the paper has focused on the QoE Controller outlining two alternative algorithms based on Reinforcement Learning concepts which have the key advantages of being model-free and of requiring a limited signalling overhead and computing power. Preliminary results (not shown for space reasons) show that the proposed solutions, based on a closed-loop, dynamic, real-time computation of the Class of Service (CoS) to be associated to each running application, seem to achieve a remarkable reduction of the QoE Error with respect to the "standard" open-loop, static policy in which the CoS are associated to the applications for their whole lifetime.

Note that the proposed dynamic approach differs from traffic classification approaches found in the literature (e.g., [12] and references in [13]), based on host-level communication behaviour-based approaches, or on statistical approaches relying on data mining methodologies, since they statically determine the CoS of the application.

Of course, the proposed approaches need an huge work to be actually designed and implemented, which is expected to be performed in the next decade.

In this respect, this paper does not propose any ready to use solution for the very complex problems related to Future Internet and, in particular, to QoE Management, but just some hints which could help in driving advanced research and future work in the Future Internet areas.

In particular, the paper has highlighted that a large amount of research work is expected especially in the field of control based and machine learning algorithms in order to design and to tune to the particular environments the QoE Evaluator and the QoE Controller algorithms (Outer Control Loop), as well as the Control functionalities and Context Engine algorithms (Inner Control Loop).

In addition, huge engineering work is also required in order to tailor the Future Internet architecture to the each considered environment; this includes, for instance, the identification of the most appropriate Present Context and Driving Parameters, of the QoE Evaluation Function Structures, of techniques for collecting user feedbacks, of the mappings among the Future Internet functional entities and the network entities, etc.

References:

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

[2] PLATINO (Grant Agreement n° PON01_01007), http://www.progettoplatino.it/.

[3] R. Jain, “Quality of experience,” IEEE Multimedia, vol. 11 , no. 1, pp. 96-95, 2004, doi: 10.1109/MMUL.2004.1261114.

[4] F. Delli Priscoli, “A Fully Cognitive Approach for Future Internet,” Molecular Diversity Preservation International - Special Issue on "Future Network Architectures" of "Future Internet", vol. 2, no. 1, pp. 16-29, 2010.

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

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

[7] S. Jelassi, G. Rubino, H. Melvin, H. Youssef, and G. Pujolle, “Quality of Experience of VoIP Service: A Survey of

Assessment Approaches and Open Issues," IEEE Communications Surveys & Tutorials, vol. 14, no. 2, pp. 491.513, 2012.

[8] P. Goudarzi, and M.R. Nezami Ranjbar, "Bandwidth allocation for video transmission with differentiated quality of experience over wireless networks," Computers and Electrical Engineering, vol. 37, no. 1, pp. 75-90, 2011.

[9] S. Singh, J.G. Andrews, and G. de Veciana, "Interference Shaping for Improved Quality of Experience for Real-Time Video Streaming," IEEE Journal on Selected Areas of Communications, vol. 30, no. 7, pp. 1259-1269, 2012.

[10] M. Fiedler, T. Hossfeld, T.-G. Phuoc "A generic quantitative relationship between quality of experience and quality of service," IEEE Network, vol. 24, no. 2, pp. 36-41, 2010.

[11] F. Delli Priscoli, M. Iannone, A. Pietrabissa, V. Suraci, "Modelling Quality of Experience in Future Internet Networks", Future Network & Mobile Summit 2012, Berlin, pp. 1-9, July 2012.

[12] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in ACM SIGCOMM, pp. 229-240, August 2005.

[13] S. Lee, H. Kim, D. Barman, C.-K. Kim, T.T. Kwon, Y. Choi, "NeTraMark: A Network Traffic classification benchmark," ACM SIGCOMM Computer Communication Review, vol. 41, no. 1, 2011.

[14] 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.

[15] 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

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

[17] D. Macone, G. Oddi, A. Palo, V. Suraci, "A common open interface to programmatically control and supervise open networks in the future internet,", in 2013 Future Network and Mobile Summit, ISBN: 978-190582437-3, Lisbon, July 2013.

[18] A. Pietrabissa, "A Reinforcement Learning Approach to Call Admission and Call Dropping Control in Links with Variable Capacity", European Journal of Control, vol. 17, no. 1, pp. 89-103, 2011, ISSN 0974-3580, DOI: 10.3166/EJC.17.89-103.

[19] 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, 2012, doi: 10.1002/ett.2536.

[20] F. Delli Priscoli, A. Isidori, "A control-engineering approach to integrated congestion control and scheduling in wireless local area networks", Control Engineering Practice, vol. 13, no. 5, pp. 541-558, 2005, doi: 10.1016/j.conengprac.2004.04.016.

[21] F. Delli Priscoli, A. Pietrabissa, "Design of a bandwidth-on-demand (BoD) protocol for satellite networks modeled as time-delay systems," Automatica, vol. 40, no. 5, pp. 729-741, May 2004, DOI:10.1016/j.automatica.2003.12.013.

[22] 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, vol. 3, pp. 46-56, June 2002.

[23] F. Delli Priscoli, "A Control Based Solution for Integrated Dynamic Capacity Assignment, Congestion Control and Scheduling in Wireless Networks," European Journal of Control, vol. 2, pp. 169-184, 2010.

[24] 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.

[25] A. Pietrabissa, "Admission Control in UMTS Networks based on Approximate Dynamic Programming," European Journal of Control, vol. 14, no. 1, pp. 62-75 , January 2008, doi:10.3166/ejc.14.62-75.

[26] A. Pietrabissa, "An Alternative LP Formulation of the Admission Control Problem in Multi-Class Networks," IEEE Transaction on Automatic Control, vol. 53, no. 3, pp. 839-845, April 2008, doi: 10.1109/TAC.2008.919516.

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

[28] F. Delli Priscoli, L. Fogliati, A. Palo, A. Pietrabissa, "Dynamic Class of Service Mapping for Quality of Experience Control in Future Networks," World Telecommunication Congress (WTC), Berlin, June 2014.

[29] R. Sutton, A. Barto, "Reinforcement Learning: An Introduction," IEEE Transactions on Neural Networks, vol. 9, pp. 1054–1054, 1998, doi:10.1109/TNN.1998.712192.

[30] C.J.H. Watkins, "Q-learning," Machine learning, vol. 8, no. 3-4, 279-292, 1992.

[31] M.L. Littman, "Friend-or-foe Q-learning in general-sum games, " ICML, vol. 1, 2001.

[32] H. Junling, and M.P. Wellman "Multiagent reinforcement learning: theoretical framework and an algorithm," ICML, vol. 98, 1998.

[33] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service", IEEE Network, vol. 24, no. 2, pp. 36-41, 2010.

[34] G. Oddi, A. Pietrabissa, "A distributed multi-path algorithm for wireless ad-hoc networks based on Wardrop routing," Proc. 21st Mediterranean Conference on Control and Automation (MED 2013), Platanias-Chania, Crete, Greece, pp. 930-935, June 25-28, 2013, doi: 10.1109/MED.2013.6608833.

[35] D. Macone, G. Oddi, A. Palo, V. Suraci, "A Dynamic Load Balancing Algorithm for Quality of Service and Mobility Management in Next Generation Home Networks," vol. 53, no. 3, 265-283, 2013, DOI 10.1007/s11235-013-9697-y.

[36] S. Manfredi, "A Reliable and Energy Efficient Cooperative Routing Algorithm for Wireless Monitoring Systems," IET Wireless Sensor Systems, vol. 2, pp. 128-135, 2012.

[37] A. Di Giorgio, L. Pimpinella, F. Liberati, "A model predictive control approach to the load shifting problem in a household equipped with an energy storage unit," 20th Mediterranean Conference on Control & Automation, Barcelona, ES, pp. 1491-1498, July 2012, doi: 10.1109/MED.2012.6265850.

[38] A. Di Giorgio A., L. Pimpinella, A. Quaresima, S. Curti, "An event driven smart home controller enabling cost effective use of electric energy and automated demand side management," 19th Mediterranean Conference on Control & Automation, Corfu, pp. 358–364, June 2011, doi: 10.1109/MED.2011.5983014.

[39] A. Di Giorgio, F. Liberati, S. Canale, "IEC 61851 compliant electric vehicle charging control in Smartgrids," 21th Mediterranean Conference on Control & Automation, Chania, GR, pp. 1329-1335, June 2013, doi: 10.1109/MED.2013.6608892.

[40] A. Di Giorgio, F. Liberati, A. Pietrabissa, "On-board stochastic control of Electric Vehicle recharging", 52nd IEEE Conference on Decision and Control, Florence (IT), pp. 5710-5715, 2013, doi: 10.1109/CDC.2013.6760789.

[41] S. Mignanti, A. Di Giorgio, V. Suraci, "A Model Based RL Admission Control Algorithm for Next Generation Networks," 2nd International Conference on Next Generation Mobile Applications, Services, and Technologies, NGMAST, Cardiff, WALES, SEP 16-19, 2008, doi: 10.1109/NGMAST.2008.19

[42] G. Oddi A. Pietrabissa, F. Delli Priscoli, V. Suraci, "A decentralized load balancing algorithm for heterogeneous wireless access networks", World Telecommunication Congress (WTC), Berlin, June 2014.

[43] P. Capodieci et al., "Improving Resilience of Interdependent Critical Infrastructures via an On-Line Alerting System," in Complexity in Engineering, Roma, pp. 88-90, 22-24 Febbraio 2010.

[44] V. Suraci, S. Mignanti, A. Aiuto, "Context-aware Semantic Service Discovery," in Proc. of Mobile and Wireless Communications Summit, Budapest, pp. 1-

5, July 2007, doi: 10.1109/ISTMWC.2007.4299110.

[45] D. Macone, G. Oddi, A. Palo, V. Suraci, “A common open interface to programmatically control and supervise open networks in the future internet”, in 2013 Future Network and Mobile Summit, Lisbon, July 2013, ISBN: 978-190582437-3.

[46] V. Suraci, A. Marucci, R. Bedini, L. Zuccaro, A. Palo,.”Energy-aware control of home networks,” Lecture Notes in Computer Science , vol. 7972, pp. 299-311, June 2013, doi: 10.1007/978-3-642-39643-4_23.

[47] ITU-T Recommendation P.862, “Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs” November 2005.

[48] J. Domingue et al., "The Future Internet", Lecture Notes in Computer Science, Volume 6656 2011, ISBN: 978-3-642-20897-3L.

[49] B. Soniu, R. Babuska and B. De Schutter, “A comprehensive survey of multi-agent reinforcement learning,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[50] R. Pecori et all, "A PKI-free Key Agreement Protocol for P2P VoIP Applications" IEEE ICC 2012, 1st International Workshop on Security and Forensics in Communication Systems, June 15th, 2012, Ottawa, Canada.

[51] R. Pecori et all, “A Statistical Blind Technique for Packet-level Recognition of IP Traffic with Dependence Enforcement” 10th IWCMC, 5th International Workshop on TRaffic Analysis and Characterization, TRAC 2014, Nicosia, Cyprus 4th-8th August, 2014.