Self-tuning of service priority parameters for optimizing Quality of Experience in LTE

P. Oliver-Balsalobre*, M. Toril*, S. Luna-Ramírez*, R. García †
*Dpt. Ingeniería de Comunicaciones
ETSI Telecomunicación, Universidad de Málaga, Campus Universitario de Teatinos, s/n
E-29071 Málaga, Spain.
Email: {pob,mtoril,sluna}@ic.uma.es
† Ericsson, Severo Ochoa, 51, E-29590 Málaga, Spain.
Email: rafael.garcia@ericsson.com

***Abstract*—Rising user expectations are changing the way mobile operators manage their networks. In this paper, a self-tuning algorithm for adjusting parameters in a multi-service packet scheduler of a Long Term Evolution (LTE) base station is proposed to optimize the overall system Quality of Experience (QoE) based on network performance statistics. For this purpose, the algorithm iteratively changes service priority parameters to re-prioritize services so as to make the most of available resources. The proposed algorithm ensures that the best overall system QoE is always reached by analyzing optimality conditions, unlike previous works, which only guarantee a minimum user satisfaction level or aim to balance QoE among services. Method assessment is carried out with a dynamic system-level simulator in a realistic service scenario. Simulation results show that the overall network QoE can be improved by up to 35% by tuning service priority parameters.**

***Index Terms*—Quality of Experience, Long Term Evolution, Self-Organizing Network, optimization, re-prioritization, services.**

I. Introduction

In years to come, an exponential growth in mobile broadband traffic is foreseen. It is expected that, from 2015 to 2021, mobile traffic will be multiplied by 10, reaching 25% of Internet total traffic [1]. At the same time, continuous advances in smartphones and tablets are modifying the general profile of services demanded by mobile users. In the past, mobile services only required very high data rates. Nowadays, most applications also require a permanent connection, significantly increasing network signaling load [2]. All these changes have made mobile network management a very challenging task [3]. In parallel, the success of mobile services has made that customers have greater choice for devices, services and applications. In a competitive market where most network and service providers offer similar products, user experience (a.k.a. Quality of Experience, QoE) has become the key differentiating factor between companies. For this reason, Customer Experience Management (CEM) has been incorporated into the daily routine of operators [4].

QoE characterization has received a great deal of attention by the mobile industry. In live networks, user experience is usually estimated from Quality of Service (QoS) metrics collected by the network, since end-to-end performance indicators are seldom available. Then, QoE modeling relates QoS and QoE indicators by means of simple analytical functions [5][6]. First QoE models for mobile services (e.g., voice [7], video-telephony [8] or video-streaming [9]) were built based on very simple network performance indicators. As vendor equipment evolves, more indicators are available for the construction of new QoE models [10][11]. More advanced QoE models also consider the impact of context on user quality perception [12][13]. Thus, two mobile users with the same QoS level might perceive very different QoE due to context features, such as user location, device technology, time of day or age.

From QoS measurements, QoE control procedures assign network resources to ensure an adequate user experience. To cope with the diversity of service requirements, the 3rd Generation Partnership Project (3GPP) has defined the QoS Class Identifier (QCI) to discriminate among different service classes [14]. Based on QCI, dynamic Packet Scheduling (PS) algorithms assign radio resources (i.e., power, frequency and time slots) based on QoS thresholds [15][16][17]. Basic schedulers provide differentiated services, QoS and user fairness by assigning appropriate weights to each user queue. More sophisticated schedulers exploit multiuser diversity gain to achieve optimal system performance [18][19][20]. In [19], a scheduling algorithm is proposed to deal with real-time and non-real time traffic in a proportional fair manner. More recent works [21][22] propose QoE-aware schedulers that ensure a minimum QoE for all users. All these schedulers decide the exact resources assigned to every single user in real time, which makes them suitable for minimum QoS/QoE assurance. However, the aim of most schedulers is to ensure a minimum satisfaction level for the worst users, rather than maximizing the overall system QoE. Moreover, implementing new advanced schedulers would require upgrading network equipment, which is not desired by network operators.

Alternatively, the overall system QoE can also be improved by tuning parameters in an existing scheduler. In [23], a self-tuning algorithm for the contention window parameter in IEEE 802.11 WLANs is presented. Such an algorithm does not discriminate between services. Similarly, an adaptation scheme is proposed in [24] for adjusting service priorities to comply with end-to-end delay constraints. In [25], another adaptive controller is proposed to enforce a maximum delay constraint for multimedia services by adjusting flow priorities. In that work, decisions are made per service, as each flow has its own controller independent from the others. Thus, the aim of each controller is to ensure that every individual traffic flow reaches a target QoS, but neglecting the requirements of other flows, which could lead to system instability. In a previous work [26], a self-tuning algorithm for a multi-service scheduler in LTE is proposed to equalize QoE among services in the context of a Self-Organizing Network (SON). In that work, QoE balance is ensured by tuning service priorities, increasing the priority of services with lower QoE. However, QoE balance does not necessarily lead to the maximum overall system QoE. To the authors' knowledge, no method has been proposed to adjust service priority parameters in a multi-service and multi-user scheduler of a radio base station so that optimal overall system QoE is guaranteed under all traffic load conditions.

In this paper, a self-tuning algorithm for adjusting service priority parameters in a classical multi-service multi-user packet scheduler of a LTE base station is proposed. Similarly to [26], the algorithm iteratively changes service priority parameters to re-prioritize services so as to make the most of available resources. Unlike in [26], the aim of the algorithm proposed here is to optimize the overall system QoE based on performance statistics in the network management system. For this purpose, an optimality condition is explicitly derived, which is then used to design the controller that ensures the optimal overall system QoE. In the problem formulation, context-aware QoE management is considered by reflecting the impact of user location on QoE perception. For this purpose, two different QoE models are provided for outdoor and indoor users. Method assessment is carried out in a dynamic system-level LTE simulator implementing a regular macrocellular scenario.

The main contributions of this work are: a) the formulation of an optimality condition for the global QoE when re-prioritizing services, b) the design of a self-tuning algorithm for adjusting service priority parameters in a LTE scheduler to maximize the global QoE, and c) a performance analysis of the proposed controller in a realistic multi-service LTE scenario.

The rest of the paper is organized as follows. Section II formulates the problem of optimizing the overall QoE in a mobile network by re-prioritizing services with scheduler parameters. Section III presents the proposed self-tuning algorithm to optimize the overall system QoE. Section IV describes simulation results and Section V presents the main conclusions of the work.

II. Problem Formulation

In this section, the system model is first outlined, including service models, scheduling algorithm and utility functions. Then, the tuning of service priority parameters is formulated as an optimization problem.

A. Service models

Service models are identical to those presented in [26]. Four services are considered:

- Voice over Internet Protocol (VoIP) is a conversational real-time service generating packets of 20 bytes every 10 ms, i.e., a bit rate of 16 kbps. A VoIP call is dropped when a user does not receive enough resources for one second.
- VIDEO service is a buffered videostreaming service where the packet arrival process is taken from real H.264/MPEG-4 AVC file traces. The load of player's buffer at the client side varies with available bandwidth and video bit rate.
- WEB service is a Hypertext Transfer Protocol (HTTP) web browsing service. A WEB session consists in a number of web pages separated by a reading time. The number of pages per session, pages size and reading time are derived from probabilistic models [27].
- FTP (File Transfer Protocol) service is a file download service. FTP session time is determined by the time spent downloading the file. As an FTP user tries to download the file at full speed, FTP session time depends on the amount of resources received during the download.

B. Scheduling algorithm

The considered PS algorithm is implemented based on the classical exponential/proportional fair (EXP/PF) scheme [18], designed for the provision of real-time services with different QoS together with non-real time services. That scheduler is extended in [26] with service priority parameters to allow re-prioritization of services. The reader is referred to [26] for more details on the scheduling algorithm.

C. Utility functions

Utility functions are closed-form expressions used to quantify user experience (i.e., QoE) from selected QoS network statistics. Due to their different nature, each service has its own utility function. In this work, context information is also considered by differentiating between indoor and outdoor locations. Thus, two utility functions are defined per service, depending on user location, $QoE^{(j_c)}$, where $j \in$ {VoIP, VIDEO, FTP, WEB} and $c \in$ {outdoor, indoor}. In all utility functions, the output is a Mean Opinion Score (MOS) value, ranging from 1 (worst) to 5 (best). To make analysis easier, indoor utility functions are obtained by changing QoS thresholds in the utility functions reported in the literature (associated here with outdoor users). Such changes are introduced to force that indoor users are more demanding in terms of QoS.

The VoIP utility functions are [28]

$$\begin{aligned} QoE^{(VoIP_{outdoor})} &= 1 + 0.035R \\ &+ 7 \times 10^{-6} R(R-60)(100-R) \ , \end{aligned} \tag{1}$$

$$\begin{aligned} QoE^{(VoIP_{indoor})} &= 1 + 0.035\tfrac{R}{1.5} \\ &+ 7 \times 10^{-6} \tfrac{R}{1.5}(\tfrac{R}{1.5}-60)(100-\tfrac{R}{1.5}) \ , \end{aligned} \tag{2}$$

where R is a factor related to traffic delay (i.e., $R = 0$ when highest delays are experienced, and $R = 100$ for the lowest values). From (1) and (2), it is deduced that $QoE^{(VoIP_{outdoor})}$ and $QoE^{(VoIP_{indoor})}$ are upper limited to 4.5 and 3.44, respectively, due to the fact that, even for ideal test conditions, some users may not rank service experience as flawless.

For buffered videostreaming services, such as YouTube or Netflix, QoE depends on how long the video takes to start (initial buffering time) and how many times and for how long the video is frozen (known as *stalling* or *re-buffering* event). Thus, videostreaming utility functions are [9]

$$\begin{aligned} QoE^{(VIDEO_{outdoor})} &= 4.23 - 0.0672\, T_{init} \\ &-0.742\, F_{reb} - 0.106\, T_{reb} \ , \end{aligned} \tag{3}$$

$$\begin{aligned} QoE^{(VIDEO_{indoor})} &= 4.23 - 0.0672\,(1.5\, T_{init}) \\ &-0.742\,(1.5\, F_{reb}) - 0.106\,(1.5\, T_{reb}) \ , \end{aligned} \tag{4}$$

where T_{init} stands for the initial buffering time in seconds, F_{reb} is the average frequency of re-buffering events in times per second, and T_{reb} is the total re-buffering time during video reproduction in seconds [9]. As in the VoIP case, the model is upper limited to a value lower than 5 (i.e., $\max(QoE^{(VIDEO_{outdoor})}) = \max(QoE^{(VIDEO_{indoor})}) = 4.23$).

The utility functions of FTP and WEB services are both based on user data throughput, as [29]

$$QoE^{(FTP_{outdoor})} = \max(1, \min(5, 0.0065\, T - 0.54)) \ , \tag{5}$$

$$QoE^{(FTP_{indoor})} = \max(1, \min(5, 0.0065\frac{T}{1.5} - 0.54)) \ , \tag{6}$$

$$QoE^{(WEB_{outdoor})} = 5 - \frac{578}{1 + \left(\frac{T+541.1}{45.98}\right)^2} \ , \tag{7}$$

$$QoE^{(WEB_{indoor})} = 5 - \frac{578}{1 + \left(\frac{\frac{T}{1.5}+541.1}{45.98}\right)^2} \ , \tag{8}$$

where T is the average user throughput in kbps. By comparing WEB and FTP functions, it is deduced that WEB is not as demanding as FTP. On the one hand, $QoE^{(WEB_{outdoor})} > QoE^{(FTP_{outdoor})}$ for low T values, showing that a WEB user needs fewer resources to perceive an acceptable service level. On the other hand, $QoE^{(WEB_{outdoor})} < QoE^{(FTP_{outdoor})}$ for high T values, so that, from some value on, higher throughput does not influence web user satisfaction, but does influence on an FTP user.

D. Optimization problem

The problem of tuning service priority parameters in the scheduler of a base station can be formulated as a classical optimization problem.

The decision variables are the service priority parameters, hereafter denoted as $SPI^{(j)}$, $j \in$ {VoIP, VIDEO, FTP, WEB}. The Figure of Merit (FoM) to be maximized is the overall system QoE, QoE_{global}, defined as

$$QoE_{global} = \frac{1}{N_s} \sum_{j=1}^{N_s} \overline{QoE}^{(j)} \ , \tag{9}$$

where N_s is the number of services in the network (4, in this work), and $\overline{QoE}^{(j)}$ is the average QoE for users of service j, calculated as

$$\overline{QoE}^{(j)} = \frac{1}{N_j} \sum_{u=1}^{N_j} QoE^{(j)}(u) \ , \tag{10}$$

where N_j is the number of users for service j and $QoE^{(j)}(u)$ is the QoE perceived by user u of service j, estimated from utility functions in (1)-(8).

In the previous formulas, the dependence of FoM (overall system QoE) on decision variables (SPI settings) is omitted for the sake of clarity. Fig. 1 breaks down all the terms involved in such a relationship. On the left of the figure, network performance depends on uncontrolled inputs (e.g., service user demand, user locations, ...) and decision variables (i.e., $SPI^{(j)}$).

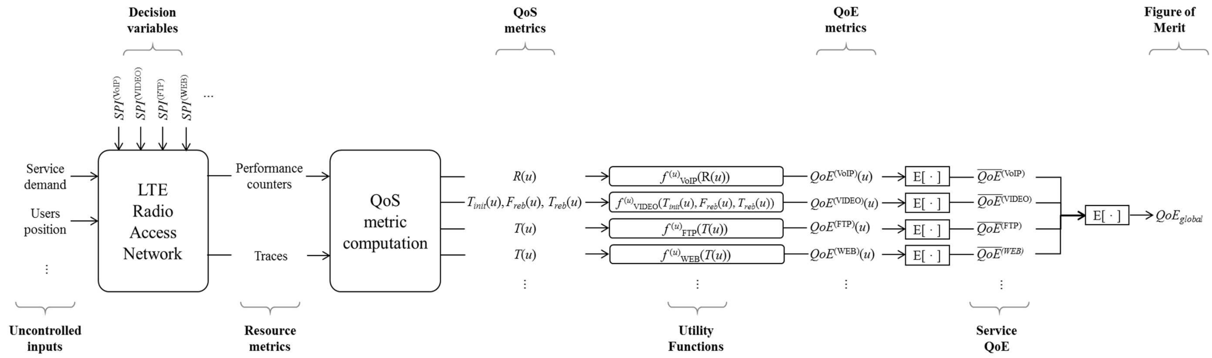


Fig. 1. Optimization problem.

Thus, increasing the SPI of a service enforces that more radio resources are assigned to users of that service, improving their QoS. Then, QoS metrics are calculated per service from network resource metrics (e.g., delay statistics, stalling statistics, ...). Later, QoE measurements are calculated per user by selecting the utility function corresponding to the service and context of user u, $f^{(u)}$, among (1)-(8). Then, service QoE is obtained by averaging QoE metrics of users of a service, including both indoor and outdoor users, as in (10). Note that context information is only used for QoE evaluation, and does not have an influence on resource or QoS metrics. Finally, the FoM of the optimization problem, QoE_{global}, is computed as in (9).

The optimization problem has two constraints. On the one hand, the SPI value is limited from 1 to 15, so that maximum and minimum service priorities are limited. On the other hand, the number of radio resources to be shared by users are limited. Thus, a high SPI value assigned to a user generating a large traffic demand will lead to a fast radio resource consumption. The former constraint has to be taken into account by the optimization algorithm, whereas the latter is already included in the QoS metrics.

The optimality conditions can be derived by considering the tuning problem as an optimization problem. Thus, the optimal configuration of SPI parameters must satisfy the stationarity condition,

$$\frac{\partial QoE_{global}}{\partial SPI^{(j)}} = 0 \ , \tag{11}$$

(i.e., the gradient of the objective function, QoE_{global}, with respect to the decision variables, $SPI^{(j)}$, is 0) for all services j where the SPI constraint is inactive (i.e., SPI has not reached its limits). For services that have reached their maximum or minimum SPI value, that decision variable is fixed to the corresponding limit, so that

$$\frac{\partial QoE_{global}}{\partial SPI^{(k)}} < 0 < \frac{\partial QoE_{global}}{\partial SPI^{(l)}} \ , \tag{12}$$

where k denotes services with $SPI = 1$ and l denotes services with $SPI = 15$.

The previous condition may give a local maximum or minimum for QoE_{global}. In practice, the convex shape of utility functions, where QoE improvements from assigning more resources gradually decrease as QoE levels increase, ensures that the above condition is only satisfied by the global optimum.

III. SELF-TUNING ALGORITHM

In this section, a self-tuning algorithm is proposed to improve the overall network QoE by changing SPI parameters in the eNodeB scheduler. The aim of tuning is to re-prioritize services so that the overall system QoE is maximized.

The algorithm is designed as a rule-based controller that iteratively modifies SPI parameters, $SPI^{(j)}$. Unlike typical controllers, based on heuristic rules, the controller proposed here is designed to drive the system to its optimal state. For this purpose, the algorithm makes use of the optimality conditions derived in the previous section.

From (11), it is deduced that the optimal controller should take the system to the equilibrium point where any change in SPI settings would deteriorate the overall QoE. This is achieved by equalizing the partial derivatives (i.e., slopes) of the overall QoE with respect to the SPI of the different services. In normal cases, such an equalization can be performed by increasing (decreasing) the SPI value of services with larger (smaller) value of partial derivative, since this action decreases (increases) the slope of the QoE_{global} function due to the convexity of utility functions.

For the above purpose, the proposed self-tuning algorithm is designed as a set of four proportional controllers (1 per service). Each controller has an incremental structure that iteratively computes the change of the SPI setting of a service from measurements of all other services. The input of the controller is the difference of the partial derivative of a service against the others, SD (for Slope Difference),

$$SD^{(j)} = \frac{\partial QoE_{global}}{\partial SPI^{(j)}} - \frac{1}{N_s - 1} \sum_{k \neq j} \frac{\partial QoE_{global}}{\partial SPI^{(k)}} \ , \quad (13)$$

i.e., the slope of QoE_{global} with respect to SPI of service j, $SPI^{(j)}$, minus the average of that of the other services. A positive value of $SD^{(j)}$ indicates that $\frac{\partial QoE_{global}}{\partial SPI^{(j)}}$ is larger than the average of $\frac{\partial QoE_{global}}{\partial SPI^{(k)}}$, so that $SPI^{(j)}$ should be increased. In contrast, a negative value of $SD^{(j)}$ indicates that $\frac{\partial QoE_{global}}{\partial SPI^{(j)}}$ is smaller than the average of $\frac{\partial QoE_{global}}{\partial SPI^{(k)}}$, meaning that $SPI^{(j)}$ should be decreased. For brevity, the expressions of partial derivatives for different services are described in Appendix A.

The output of the controller is the positive or negative step to be added to the current SPI value. Fig. 2 shows the input-output relation in one of the controllers. In each iteration, the SPI parameter change, $\Delta SPI^{(j)}$, is computed from the slope difference, $SD^{(j)}$. Basically, $SPI^{(j)}$ is increased/decreased so as to reduce the slope difference against other services, as this should increase QoE_{global}. A piecewise function with five segments and three different slopes is selected to implement a gain scheduling algorithm, providing adequate trade-off between speed of response and system stability. Segments with a higher slope aim to increase convergence speed towards the balance point at the start of the equalization process (i.e., when $SD^{(j)}$ is still large). Segments on the sides with a zero slope ensure system stability by limiting the maximum value of $\Delta SPI^{(j)}$ to ± 2. Likewise, the central segment with the lower slope ensures system stability at the end of the balancing process (i.e., when $SD^{(j)}$ is small).

The pseudocode of the algorithm is shown in Fig. 3. Both the input and the output of the algorithm are detailed at the top. The algorithm is run a pre-determined number of times (optimization loops). First, the algorithm computes the partial derivatives required to generate the input of the controller, $SD^{(j)}$. Then, the output of the controller, $\Delta SPI^{(j)}$, is obtained from $SD^{(j)}$ based on the function in Fig. 2. Finally, the new parameter value, $SPI^{(j)\prime}$, is obtained by adding the step to the current value, $SPI^{(j)}$, and enforcing that the value is between 1 and 15.

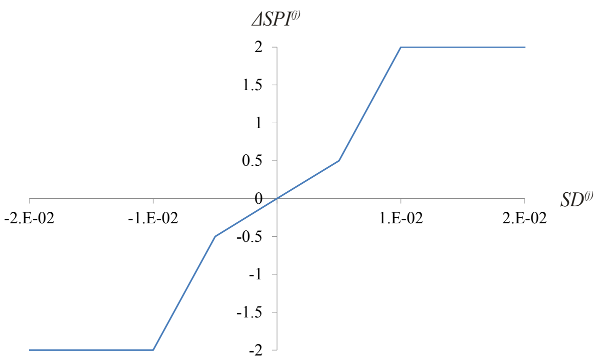


Fig. 2. Input-output relation in the proposed controller.

Note that the number of controllers is equal to the number of services (i.e., one per service for the whole network), and SPI tuning is therefore done on a network basis (i.e., all eNodeBs in the network share the same SPI configuration). The four controllers work independently, but synchronously, sharing network performance measurements from the past iteration and computing SPI changes in their service for the next iteration.

IV. Performance assessment

For clarity, the assessment methodology is first described and results are presented later.

A. Assessment methodology

A dynamic system-level LTE simulator is used [30]. A regular macrocellular scenario is considered, consisting of 19 tri-sectorized sites uniformly distributed [26]. To reduce the computational load, the minimum system bandwidth is set (i.e., 6 Physical Resource Blocks, PRBs) and only the downlink is simulated. For ease of analysis, uniform user spatial distribution is considered. The scheduling algorithm includes service priority parameters (i.e., SPI), as explained in Section II-B. The reader is referred to [30] for additional details about the simulation tool.

To check the impact of service distribution, two realistic traffic mixes are considered, referred to as *Traffic mix A* and *B*. As shown in Table I [27], VoIP is the most populated service in both distributions. The main difference is the share of video users, which are those demanding more resources. As a result, offered traffic in mix A is dominated by video users, while offered traffic in mix B is more evenly balanced. Unless stated otherwise, traffic mix A is used. Service models were described in Section II-A. Network load is controlled by the average number of users per cell. In both traffic mixes, this parameter is fixed to ensure that PRB utilization ratio is close to 100% for every single cell, so that services compete for available radio resources. Thus, SPI should have a strong impact on user experience. Such a load configuration sets a worst-case scenario in terms of inter-cell interference.

The proposed self-tuning algorithm (referred to as *optimization* algorithm) is compared with the simple QoE balancing algorithm in [26] (referred to as *balancing* algorithm). Both schemes consist of 4 controllers (i.e., VoIP, VIDEO, WEB, FTP), but the former intends to achieve optimal system performance, whereas the latter just equalizes the QoE among services.

Five experiments are carried out. The first four experiments only consider outdoor users, while the

Input: D, R, T_{reb}, T_{WEB}, T_{FTP}

N_{VoIP}, N_{VIDEO}, N_{WEB}, N_{FTP}, N_s

$PRB_{util}^{(VoIP)}$, $PRB_{util}^{(VIDEO)}$, $PRB_{util}^{(WEB)}$, $PRB_{util}^{(FTP)}$

$SPI^{(VoIP)}$, $SPI^{(VIDEO)}$, $SPI^{(WEB)}$, $SPI^{(FTP)}$

Output: $SPI^{(VoIP)\prime}$, $SPI^{(VIDEO)\prime}$, $SPI^{(WEB)\prime}$, $SPI^{(FTP)\prime}$

For a pre-determined number of optimization loops

For every service j in the network, with k representing other services ($k \neq j$)

$$\frac{\partial QoE^{(j)}}{\partial SPI^{(j)}} = \frac{\partial QoE^{(j)}}{\partial QoS^{(j)}} \cdot \frac{\partial QoS^{(j)}}{\partial Res^{(j)}} \cdot \frac{\partial Res^{(j)}}{\partial SPI^{(j)}}$$

$$\frac{\partial QoE^{(j)}}{\partial QoS^{(j)}} = \frac{1}{N_j} \sum_{u=1}^{N_j} \frac{\partial QoE^{(j)}}{\partial QoS^{(j)}}(u)$$

$$\frac{\partial QoS^{(j)}}{\partial Res^{(j)}} \simeq \frac{QoS_{max}^{(j)} - QoS_{min}^{(j)}}{Res_{max}^{(j)} - Res_{min}^{(j)}}$$

$$Res_{max}^{(j)} = (SPI_{max} - SPI_{min}) \sum_{\forall k \neq j} PRB_{util}^{(k)}$$

$$Res_{min}^{(j)} = (SPI_{min} - SPI_{max}) \sum_{\forall k \neq j} PRB_{util}^{(k)}$$

$$\frac{\partial Res^{(j)}}{\partial SPI^{(j)}} \simeq \sum_{\forall\, k \neq j} PRB_{util}^{(k)}$$

$$\frac{\partial QoE^{(k)}}{\partial SPI^{(j)}} = \frac{\partial QoE^{(k)}}{\partial QoS^{(k)}} \cdot \frac{\partial QoS^{(k)}}{\partial Res^{(k)}} \cdot \frac{\partial Res^{(k)}}{\partial SPI^{(j)}}$$

$$\frac{\partial QoE^{(k)}}{\partial QoS^{(k)}} = \frac{1}{N_k} \sum_{u=1}^{N_k} \frac{\partial QoE^{(k)}}{\partial QoS^{(k)}}(u)$$

$$\frac{\partial QoS^{(k)}}{\partial Res^{(k)}} \simeq \frac{QoS_{max}^{(k)} - QoS_{min}^{(k)}}{Res_{max}^{(k)} - Res_{min}^{(k)}}$$

$$Res_{max}^{(k)} = (SPI_{max} - SPI_{min}) \sum_{\forall l \neq k} PRB_{util}^{(l)}$$

$$Res_{min}^{(k)} = (SPI_{min} - SPI_{max}) \sum_{\forall l \neq k} PRB_{util}^{(l)}$$

$$\frac{\partial Res^{(k)}}{\partial SPI^{(j)}} \simeq -PRB_{util}^{(j)}$$

$$\frac{\partial QoE_{global}}{\partial SPI^{(j)}} = \frac{1}{N_s} \sum_{k=1}^{N_s} \frac{\partial QoE^{(k)}}{\partial SPI^{(j)}} = \frac{1}{N_s} \left(\frac{\partial QoE^{(j)}}{\partial SPI^{(j)}} + \sum_{k=1}^{N_s - 1} \frac{\partial QoE^{(k)}}{\partial SPI^{(j)}} \right)$$

End For

For every service j in the network, with k representing other services ($k \neq j$)

$$SD^{(j)} = \frac{\partial QoE_{global}}{\partial SPI^{(j)}} - \frac{1}{N_s - 1} \sum_{k \neq j} \frac{\partial QoE_{global}}{\partial SPI^{(k)}}$$

$$\Delta SPI^{(j)} = f\left(SD^{(j)}\right)$$

$$SPI^{(j)\prime} = \max\left(1, \min\left(15, SPI^{(j)} + \Delta SPI^{(j)}\right)\right)$$

End For

End For

Fig. 3. Pseudocode of the self-tuning algorithm.

last experiment checks the robustness of the algorithm when both outdoor and indoor users are considered.

- In a first preliminary experiment, the aim is to explain the basic performance of the algorithm by showing the value of intermediate terms that are necessary to obtain the input of the controller, $SD^{(j)}$, in a typical situation. The result of this experiment is then used as the first loop of the proposed algorithm. Readers who have not yet read the Appendix may skip this experiment.
- A second experiment shows the benefits of the new algorithm compared to the previous approach of balancing QoE among services. In this experiment, all services begin with the same SPI value (i.e., $SPI^{(j)} = 8\ \forall\, j$), i.e., no service is initially prioritized over the others. It is expected that both self-tuning algorithms change SPI on a per-service basis, but only the one proposed here achieves optimal QoE_{global}.
- A third experiment proves the robustness of the proposed algorithm by showing its capability to optimize the QoE_{global} also for non-uniform initial SPI configurations. For this purpose, a set of 10 initial SPI configurations are randomly gener-

TABLE I
TRAFFIC MIX

Service	QCI (Traffic Category)	Share of users [%]	
		Mix A	Mix B (Experiment 4)
VoIP	1 (Real-Time)	50	65
VIDEO	6 (Streaming)	20	5
WEB	8 (Interactive)	20	20
FTP	9 (Best effort)	10	10

TABLE II
$QoS^{(j)}$ VS $Res^{(j)}$ (TRAFFIC MIX A, UNIFORM INITIAL SPI)

Service (j)	$QoS^{(j)}$	$QoS^{(j)}_{max}$	$QoS^{(j)}_{min}$	$PRB^{(j)}_{util}$	$\sum_{\forall k \neq j} PRB^{(k)}_{util}$	$Res^{(j)}_{max}$	$Res^{(j)}_{min}$	$\frac{\partial QoS^{(j)}}{\partial Res^{(j)}}$
VoIP	D [s]	1	0	0.05	0.95	13.30	-13.30	-0.04
VIDEO	T_{reb} [s]	0	30	0.51	0.49	6.86	-6.86	-2.2
FTP	T [kbps]	1200	0	0.29	0.71	9.94	-9.94	60.4
WEB	T [kbps]	1200	0	0.15	0.85	11.90	-11.90	50.3

ated with a uniform parameter distribution. Then, the proposed algorithm is applied to the network with each initial settings (i.e., the algorithm is run 10 times).

- A fourth experiment checks the sensitivity of the algorithm to the service mix by simulating traffic mix B. To ease the analysis, all services begin with the same SPI value (i.e., $SPI^{(j)} = 8 \ \forall j$).
- A fifth experiment includes context information. Different utility functions are used, depending on whether the user is indoors or outdoors. For this experiment, 70% of users are created indoors [31]. All services begin with the same SPI value (i.e., $SPI^{(j)} = 8 \ \forall j$).

Each simulation includes 24 optimization loops, each consisting of 5 minutes of network time (i.e., SPI parameters are modified every 5 minutes). It is checked a posteriori that 24 loops are enough to ensure that the system reaches equilibrium in all experiments. To ensure that every optimization loop within the same experiment is carried out under identical conditions, all random variables are pre-generated. Thus, performance differences between loops are only due to SPI tuning, and not to the stochastic nature of simulation.

The main figure of merit is the overall system QoE in equilibrium (i.e., at the end of the tuning process). Stability and convergence speed are secondary criteria. In each experiment, network performance with the initial SPI settings (i.e., before changing SPI parameters) is used as a benchmark.

B. Results

1) Experiment 1: Basic performance (traffic mix A, uniform initial SPI): The aim of this experiment is to show an example of how the controllers' input, $SD^{(k)}$, is computed from network statistics. Table III summarizes the values of the three partial derivatives, $\frac{\partial QoE^{(k)}}{\partial QoS^{(k)}}$, $\frac{\partial QoS^{(k)}}{\partial Res^{(k)}}$ and $\frac{\partial Res^{(k)}}{\partial SPI^{(j)}}$, whose product is the slope of the QoE of a service with respect to the SPI change of another service, $\frac{\partial QoE^{(k)}}{\partial SPI^{(j)}}$, as shown in Eq. (16) in the Appendix. Table II breaks down the terms required to compute the second factor, $\frac{\partial QoS^{(k)}}{\partial Res^{(k)}}$, as described in Eq. (24)-(26).

The analysis is first focused on the values of $\frac{\partial QoS^{(j)}}{\partial Res^{(j)}}$ shown in Table II. Recall that term shows the QoS improvement of re-assigning resources to a service by taking them from the other services. From the sign of $\frac{\partial QoS^{(j)}}{\partial Res^{(j)}}$ in the last column, it is verified that $QoS^{(j)}$ is inversely proportional to $Res^{(j)}$ in the case of VoIP and VIDEO, since both delay and re-buffering time decrease when more resources are allocated. In contrast, $QoS^{(j)}$ increases with $Res^{(j)}$ for FTP and WEB, since a larger amount of resources leads to a higher throughput. Likewise, the large differences in the values in the last column come from the large differences in the QoS thresholds in the third and fourth columns.

A more detailed analysis is presented in Table III, which shows the values of the three factors determining the slope of the QoE of a service with respect to the changes of SPI of another service. First, it is observed that the first two factors, $\frac{\partial QoE^{(k)}}{\partial QoS^{(k)}}$ and $\frac{\partial QoS^{(k)}}{\partial Res^{(k)}}$, are vectors, as they are computed on a per-service basis, while the third factor, $\frac{\partial Res^{(k)}}{\partial SPI^{(j)}}$, is a matrix whose elements are computed for each pair of services. An inspection of the values of the second and third columns shows that the sign of the first two factors is the same. Again,

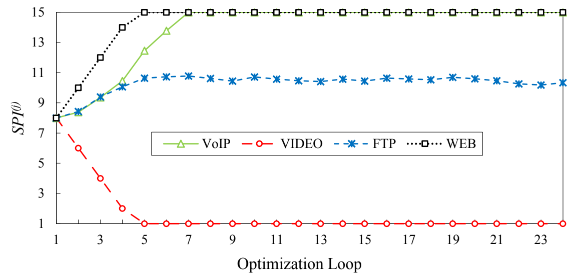
TABLE III
$QoE^{(k)}$ VS $SPI^{(j)}$ (TRAFFIC MIX A, UNIFORM INITIAL SPI)

Service (k)	$\frac{\partial QoE^{(k)}}{\partial QoS^{(k)}}$	$\frac{\partial QoS^{(k)}}{\partial Res^{(k)}}$	$\frac{\partial Res^{(k)}}{\partial SPI^{(j)}}$				$\frac{\partial QoE^{(k)}}{\partial SPI^{(j)}} = \frac{\partial QoE^{(k)}}{\partial QoS^{(k)}} \cdot \frac{\partial QoS^{(k)}}{\partial Res^{(k)}} \cdot \frac{\partial Res^{(k)}}{\partial SPI^{(j)}}$				$SD^{(k)}$	$\Delta SPI^{(k)}$
			VoIP	VIDEO	FTP	WEB	VoIP	VIDEO	FTP	WEB		
VoIP	-1.06	-0.04	0.95	-0.51	-0.29	-0.15	0.04	-0.02	-0.01	-0.01	0.004	0.39
VIDEO	-0.08	-2.2	-0.05	0.49	-0.29	-0.15	-0.01	0.09	-0.05	-0.03	-0.044	-2.00
FTP	0.003	60.4	-0.05	-0.51	0.71	-0.15	-0.01	-0.09	0.13	-0.03	0.004	0.43
WEB	0.004	50.3	-0.05	-0.51	-0.29	0.85	-0.01	-0.10	-0.06	0.17	0.036	2.00

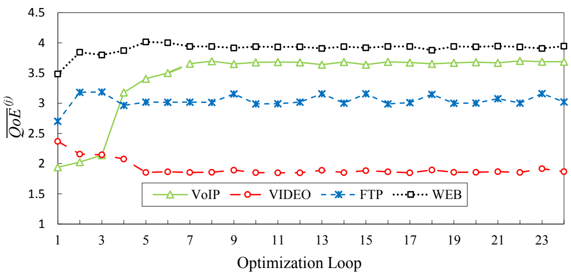
for VoIP and VIDEO, a decrease in the QoS indicator (i.e., average delay and re-buffering time) improves QoE, whereas, for FTP and WEB, it is an increase in the QoS indicator (i.e., average user throughput) that improves QoE. Regarding the $\frac{\partial QoE^{(k)}}{\partial SPI^{(j)}}$ term, note that the main diagonal ($k = j$) is always greater than 0, because increasing the SPI of a service (i.e., prioritizing a service) most often leads to an increase of its QoE. In contrast, non-diagonal elements are all negative, as increasing the SPI of other services (i.e., prioritizing other services) tends to degrade the QoE of a service. Finally, the last two columns show the slope difference, $SD^{(k)}$, and the suggested SPI change, $\Delta SPI^{(k)}$, which are the input and output of the controllers, respectively. Both parameters are related by the controller function shown in Fig. 2. From the output values, it is deduced that, in the initial loop with traffic mix A and uniform initial SPI settings, the best action to improve the overall system QoE is to increase the SPI of WEB, FTP and VoIP, and decrease the SPI of VIDEO.

2) *Experiment 2: Optimization vs balancing algorithm (traffic mix A, uniform initial SPI):* Fig. 4(a)-(b) show the evolution of $SPI^{(j)}$ and $\overline{QoE}^{(j)}$, respectively, with the proposed algorithm. Fig. 5 shows the evolution of the overall system QoE for both self-tuning algorithms. In Fig. 4, it is observed that, at the first iteration, VoIP and WEB services have very disparate QoE values ($\overline{QoE}^{(VoIP)} = 1.94$ and $\overline{QoE}^{(WEB)} = 3.5$), regardless of the uniform SPI settings. With this initial configuration, the scheduler benefits WEB by allocating enough resources to download web pages in a reasonable time (and, thus, $\overline{QoE}^{(WEB)}$ is high), but not enough for a satisfactory packet delay for VoIP service (and, thus, $\overline{QoE}^{(VoIP)}$ is low). As a consequence, $QoE_{global} = 2.62$ in the first iteration.

Fig. 4(a) shows that, as tuning is performed, all services except VIDEO increase their $SPI^{(j)}$ value. In particular, VoIP and WEB SPI values end up at the maximum value ($SPI^{(VoIP)} = SPI^{(WEB)} = 15$), while $SPI^{(FTP)}$ stabilizes around 10.5. This is achieved at the expense of VIDEO users, for which $\overline{QoE}^{(VIDEO)}$ is slightly degraded from 2.4 to 1.8 (first and last iterations in Fig. 4(b)). As a result, QoE_{global} is significantly improved from 2.62 to 3.13 (first and last iterations in Fig. 5). As explained below, this is



(a) SPI



(b) Service QoE

Fig. 4. Evolution of SPI and QoE for different services (traffic mix A, uniform initial SPI).

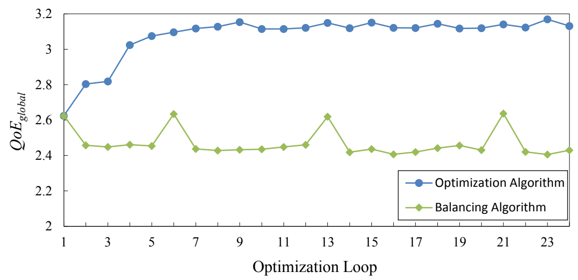


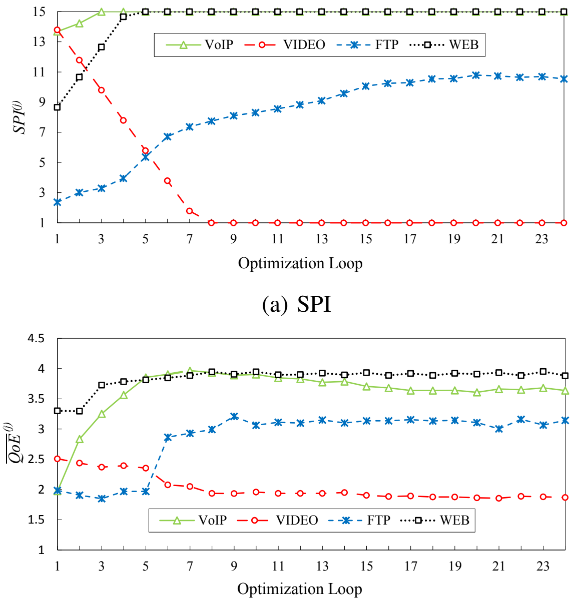
Fig. 5. Evolution of the overall system QoE for different algorithms (traffic mix A, uniform initial SPI).

just a consequence of the re-prioritization of services that leads to the re-assignment of the large amount of resources initially used by VIDEO users to other services.

The self-tuning algorithm mainly acts in the first eight iterations, with no significant changes of SPI and QoE indicators in subsequent iterations. In Fig. 5, an abrupt change in QoE_{global} is seen in the 4^{th} optimization loop. That is the point when the priority of the voice service becomes higher than FTP service priority (i.e., $SPI^{(VoIP)} > SPI^{(FTP)}$). A more detailed analysis (not shown here) reveals that QoE_{global} is considerably increased every time the SPI value of a service with a high demand of radio resources (FTP and VIDEO in this work) falls below the SPI of another service. Similar peaks are observed in the overall system QoE for the QoE balancing algorithm. More importantly, the proposed self-tuning algorithm based on the optimality condition achieves a 20% improvement in the global QoE (i.e., from 2.62 to 3.13), whereas the QoE balancing algorithm deteriorates the global QoE by 7% (i.e., from 2.62 to 2.43).

3) Experiment 3: Impact of initial SPI configuration (traffic mix A, random initial SPI): The proposed algorithm is simulated with 10 initial SPI configurations generated at random. Fig. 6 shows the SPI and QoE evolution per service with one of the initial configurations ($SPI^{(VoIP)} = 13.68$, $SPI^{(VIDEO)} = 13.79$, $SPI^{(FTP)} = 2.37$ and $SPI^{(WEB)} = 8.66$). In the example, FTP and VoIP are the services with the worst initial user experience ($\overline{QoE}^{(FTP)} = 1.98$ and $\overline{QoE}^{(VoIP)} = 1.97$). For FTP, the reason for the low initial QoE is a low initial SPI ($SPI^{(FTP)} = 2.37$). For VoIP, the reason is not the initial SPI value, but a SPI slightly lower than that of VIDEO ($SPI^{(VoIP)} = 13.68$ vs $SPI^{(VIDEO)} = 13.79$). Such a small difference with the SPI of a resource-hungry application as VIDEO makes VoIP users suffer long delays in the scheduler, resulting in a bad QoE for VoIP. Other than that, the algorithm behaves similarly to the first experiment. Fig. 6(b) show that all services, except VIDEO, improve their experience. Fig. 6(a) shows that VoIP and WEB SPI reach the maximum $SPI^{(j)}$ value (i.e., 15). Fig. 7 shows that QoE_{global} is significantly improved from 2.44 to 3.13 (28% enhancement).

A detailed analysis of the first iteration with the different initial SPI configurations (not shown here) reveals that the initial overall system QoE varies between 2.32 and 3.00 among the 10 cases. In all cases, the proposed algorithm manages to lead the system to the optimal state at the end of the tuning process, where $QoE_{global} = 3.13$. Specifically, an average improvement of 21.8% is obtained in QoE_{global} compared to the initial situation, with a minimum and



(a) SPI

(b) Service QoE

Fig. 6. Evolution of SPI and QoE for different services (traffic mix A, random initial SPI).

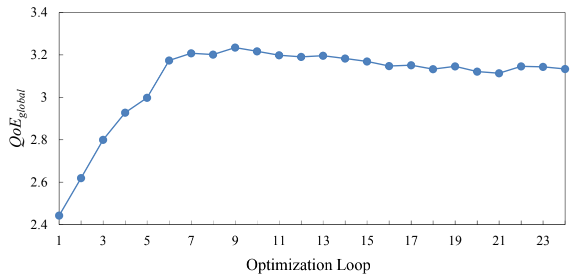
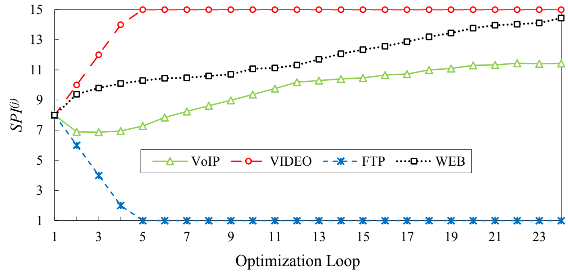


Fig. 7. Example of evolution of the overall system QoE with uneven initial SPI settings (traffic mix A, random initial SPI)

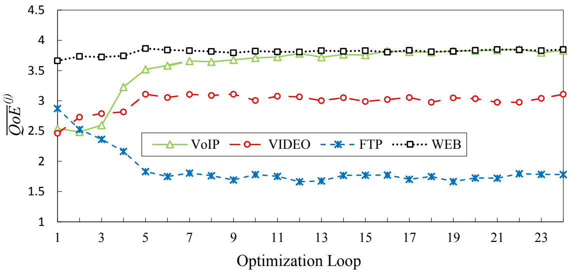
maximum value of 4.33% and 35.34%, respectively. It is worth remarking that QoE_{global} is never decreased, regardless of the initial SPI configuration. This was not the case for the QoE balancing algorithm in the previous experiment.

4) Experiment 4: Impact of traffic mix (traffic mix B, uniform initial SPI): In this experiment, a different traffic mix is configured in the scenario. As shown in Table I, traffic mix B has significantly less video users, causing that traffic demand is not dominated by this service, as in traffic mix A. For ease of analysis, initial SPI settings are configured the same for all services.

Fig. 8(a) shows how the proposed self-tuning algorithm adjusts SPI settings to the new traffic conditions. In this case, VIDEO is the service with the fastest $SPI^{(j)}$ increase, reaching the maximum value ($SPI^{(VIDEO)} = 15$) in barely 5 loops. In contrast, FTP service priority parameter is decreased to the



(a) SPI



(b) Service QoE

Fig. 8. Evolution of SPI and QoE for a different traffic mix (traffic mix B, uniform initial SPI).

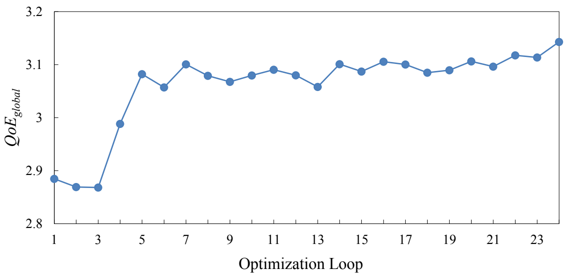
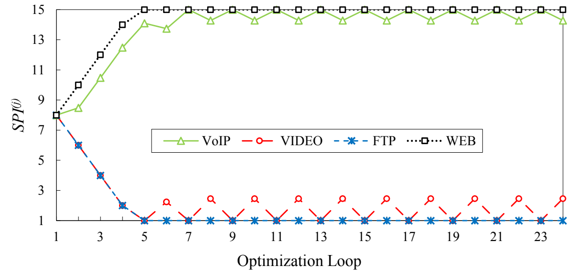
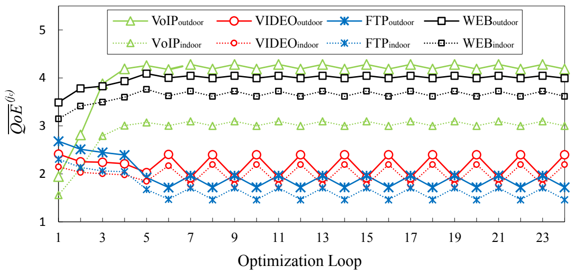


Fig. 9. Evolution of overall system QoE for a different traffic mix (traffic mix B, uniform initial SPI).



(a) SPI



(b) Service-context QoE

Fig. 10. Evolution of SPI and QoE for outdoor and indoor users (traffic mix A, uniform initial SPI)

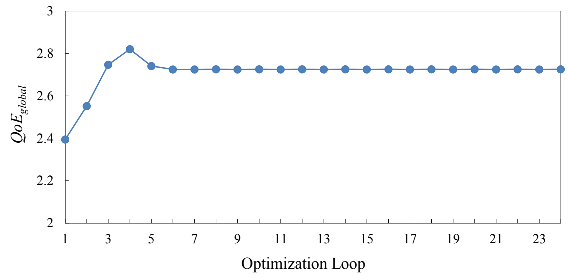


Fig. 11. Evolution of overall system QoE considering outdoor and indoor users (traffic mix A, uniform initial SPI).

minimum ($SPI^{(FTP)} = 1$). Due to the new service distribution, where FTP users are more than VIDEO users (10% against 5%), both services interchange their SPI and QoE behaviors. Thus, less frequent VIDEO users are now awarded with a higher SPI and, consequently, better QoE. More frequent FTP users are penalized with a lower SPI and worse QoE. VoIP and WEB services behave similarly to the previous experiments in terms of SPI and QoE, but with slower increases. Again, QoE_{global} is improved by 9% (from 2.88 to 3.14 at first and last iterations), as shown in Fig. 9.

5) Experiment 5: Context dependency (traffic mix A, uniform initial SPI): This experiment checks the impact of context information. Fig. 10(a)-(b) show the evolution of $SPI^{(j)}$ and $\overline{QoE}^{(j_c)}$ with the proposed algorithm. In Fig. 10(a), it can be seen how $SPI^{(VoIP)}$ and $SPI^{(WEB)}$ increase up to the maximum level in a few iterations, while VIDEO and FTP priorities decrease at a similar speed up to their minimum value. This behavior is also reflected in service-context QoE values, shown in Fig. 10(b). Differences between outdoor and indoor VoIP users in Fig. 10(b) are due to differences in indoor/outdoor utility functions, (1) and (2). Both functions depend on the R factor, although indoor function includes a correction factor (1.5). This factor leads to lower MOS values for indoor users even when a high R (i.e., low packet delay) is experienced. The system reaches steady state at the 7^{th} loop. Thereafter, some fluctuations are observed in QoE, especially in VIDEO and FTP services.

Finally, Fig. 11 illustrates QoE_{global} evolution. QoE_{global} improves significantly in the first five loops, and remains stable from that point. Specifically, FoM is improved by a 14% (i.e., from 2.39 to 2.73). This result proves that the proposed algorithm can deal with several utility functions per service for context-aware QoE modeling.

V. Conclusions

In this paper, a self-tuning algorithm for a classical multi-service packet scheduler in a LTE base station has been proposed. The aim of the algorithm is to achieve the optimum overall system QoE by modifying service priorities, regardless of network conditions. For this purpose, the proposed algorithm changes service priority parameters to re-prioritize services by an iterative process based on network statistics. Briefly, the algorithm tends to increase the priority of services that would make the most of additional resources and are occupying less resources. Method assessment has been carried out in a dynamic system-level LTE simulator. Results have shown that tuning service priority parameters can improve the overall system QoE for different initial configurations and traffic mixes, with gains of up to 35%. Such an improvement is obtained without changing the existing scheduling algorithm. Furthermore, the proposed algorithm also improves network performance when user location information is included in QoE models. It is left for future work the inclusion of other context attributes in QoE models.

The proposed self-tuning algorithm is conceived as a centralized algorithm for the network management system, where service performance indicators and priority parameters are collected and set on a network basis. However, it can also be designed as a distributed algorithm executed in every eNodeB, provided that service performance statistics are available and priority parameters can be adjusted on a site or cell basis.

Appendix A

In this Appendix, partial derivatives $\frac{\partial QoE_{global}}{\partial SPI^{(j)}}$ are developed. These terms, reflecting how the overall system QoE is affected by changes in the SPI of a service j, are used to define the controller in (13). For space reasons, only outdoor users are considered in this appendix. A similar development has been carried for indoor users to conduct the last experiment in Section IV.

From (9), those partial derivatives are computed as

$$\begin{aligned} \frac{\partial QoE_{global}}{\partial SPI^{(j)}} &= \frac{1}{N_s}\sum_{k=1}^{N_s}\frac{\partial QoE^{(k)}}{\partial SPI^{(j)}} \\ &= \frac{1}{N_s}\left(\frac{\partial QoE^{(j)}}{\partial SPI^{(j)}} + \sum_{k=1}^{N_s-1}\frac{\partial QoE^{(k)}}{\partial SPI^{(j)}}\right) \ , \end{aligned} \tag{14}$$

where N_s is the number of services. For convenience, an intermediate term is defined per service, $Res^{(k)}$, as

$$Res^{(k)} = \sum_{\forall j \neq k}(SPI^{(k)} - SPI^{(j)}) \cdot PRB_{util}^{(j)} \ , \tag{15}$$

where $PRB_{util}^{(j)}$ represents the PRB utilization ratio of service j (i.e., amount of radio resources used by service j against total amount of resources) in the current optimization loop. This intermediate term roughly reflects the maximum ratio of resources that can potentially be gained by a service from other services.

With (15), partial derivatives $\frac{\partial QoE^{(k)}}{\partial SPI^{(j)}}$, reflecting how the QoE of service k is affected by SPI changes in another service j, can be developed as

$$\frac{\partial QoE^{(k)}}{\partial SPI^{(j)}} = \frac{\partial QoE^{(k)}}{\partial QoS^{(k)}} \cdot \frac{\partial QoS^{(k)}}{\partial Res^{(k)}} \cdot \frac{\partial Res^{(k)}}{\partial SPI^{(j)}} \ , \tag{16}$$

where $QoS^{(k)}$ is the main QoS indicator in the QoE model for service k (i.e., average delay for VoIP, re-buffering time for VIDEO or throughput for WEB and FTP). The second and third factors are shared by all users of a service, and are therefore calculated on a per-service basis. In contrast, the first factor, $\frac{\partial QoE^{(k)}}{\partial QoS^{(k)}}$, representing the slope of the utility function, depends on the specific QoS level provided to each user, and could be computed on a per-user basis. As the SPI parameter is defined on a service basis, SPI changes are selected by aggregating users of a service. Thus, an average slope is computed as

$$\frac{\partial QoE^{(k)}}{\partial QoS^{(k)}} = \frac{1}{N_k}\sum_{u=1}^{N_k}\frac{\partial QoE^{(k)}}{\partial QoS^{(k)}}(u) \ , \tag{17}$$

where N_k is the number of users for service k and $\frac{\partial QoE^{(k)}}{\partial QoS^{(k)}}(u)$ is the slope for a particular user u.

A. QoE vs QoS

The first factor, $\frac{\partial QoE^{(k)}}{\partial QoS^{(k)}}$, is computed by aggregating the slopes for users of a service. Such slopes are obtained analytically on a per-user and per-iteration basis from the utility function of the service (1)-(8) as follows.

In (1), $QoE^{(VoIP)}$ depends on R, which is computed from connection delay, D, in seconds. Thus, the main QoS indicator for VoIP is delay (i.e., $QoS^{(VoIP)} = D$), and

$$\frac{\partial QoE^{(VoIP)}}{\partial QoS^{(VoIP)}}(u) = \frac{\partial QoE^{(VoIP)}}{\partial R}(u) \cdot \frac{\partial R}{\partial D}(u) \ . \tag{18}$$

From (1),

$$\begin{aligned} \frac{\partial QoE^{(VoIP)}}{\partial R}(u) &= 0.035 + 7 \times 10^{-6}(200R(u) \\ &\quad -3R(u)^2 - 6 \times 10^3 + 120R(u)) \ , \end{aligned} \tag{19}$$

and, from [28],

$$\frac{\partial R}{\partial D}(u) = -0.024 - 0.11\,\mathrm{H}(D(u) - 177.3) \quad /$$

$$\mathrm{H}(D(u) - x) = \begin{cases} 0 & D(u) < x \\ 1 & D(u) \geq x \end{cases} . \tag{20}$$

In (3), the re-buffering time component tends to dominate the QoE of VIDEO, so that $QoE^{(VIDEO)} \approx 4.23 - 0.106\,T_{reb}$. Thus, the main QoS indicator for VIDEO is the re-buffering time in seconds (i.e., $QoS^{(VIDEO)} = T_{reb}$), and

$$\frac{\partial QoE^{(VIDEO)}}{\partial QoS^{(VIDEO)}}(u) = \frac{\partial QoE^{(VIDEO)}}{\partial T_{reb}}(u)$$

$$= \begin{cases} -0.106 & T_{reb}(u) < 30 \\ 0 & T_{reb}(u) \geq 30 \end{cases} , \tag{21}$$

where $T_{reb}(u) = 30$ is the threshold from which $QoE^{(VIDEO)}$ is always minimum ($= 1$).

In (5) and (7), the main QoS indicator for FTP and WEB services is average user throughput in kbps (i.e., $QoS^{(WEB)} = QoS^{(FTP)} = T$), and

$$\frac{\partial QoE^{(WEB)}}{\partial QoS^{(WEB)}}(u) = \frac{\partial QoE^{(WEB)}}{\partial T}(u)$$

$$= \frac{578\left(\frac{1}{45.98}\right)^2 (2\,T(u)+1.082)}{\left(1+\left(\frac{T(u)+541.1}{45.98}\right)^2\right)^2} , \tag{22}$$

$$\frac{\partial QoE^{(FTP)}}{\partial QoS^{(FTP)}}(u) = \frac{\partial QoE^{(FTP)}}{\partial T}(u)$$

$$= \begin{cases} 0 & T(u) \leq 236.5 \\ 0.0065 & 236.5 < T(u) < 852.5 \\ 0 & T(u) \geq 852.5 \end{cases} . \tag{23}$$

In (23), throughput values represent the thresholds associated with the minimum ($= 1$) and maximum ($= 5$) values of the MOS scale for FTP.

B. QoS vs Res

The second factor, $\frac{\partial QoS^{(k)}}{\partial Res^{(k)}}$, is estimated on a per-iteration and per-service basis as

$$\frac{\partial QoS^{(j)}}{\partial Res^{(j)}} \simeq \frac{QoS^{(j)}_{max} - QoS^{(j)}_{min}}{Res^{(j)}_{max} - Res^{(j)}_{min}} , \tag{24}$$

where $QoS^{(j)}_{max}$ and $QoS^{(j)}_{min}$ stand for the maximum and minimum value of the main QoS indicator per service, respectively, and $Res^{(j)}_{max}$ and $Res^{(j)}_{min}$ are obtained as

$$Res^{(j)}_{max} = (SPI_{max} - SPI_{min}) \sum_{\forall k \neq j} PRB^{(k)}_{util} , \tag{25}$$

$$Res^{(j)}_{min} = (SPI_{min} - SPI_{max}) \sum_{\forall k \neq j} PRB^{(k)}_{util} . \tag{26}$$

C. Res vs SPI

The third factor, $\frac{\partial Res^{(k)}}{\partial SPI^{(j)}}$, is approximated on a per-iteration and per-service basis by

$$\frac{\partial Res^{(k)}}{\partial SPI^{(j)}} \simeq \begin{cases} -PRB^{(j)}_{util} & k \neq j \\ \sum\limits_{\forall\, l \neq k} PRB^{(l)}_{util} & k = j \end{cases} . \tag{27}$$

Acknowledgment

This work has been funded by the Spanish Ministry of Economy and Competitiveness (TEC2015-69982-R) and Optimi-Ericsson and Agencia IDEA (Consejería de Ciencia, Innovación y Empresa, Junta de Andalucía, ref. 59288), co-funded by FEDER.

References

[1] Ericsson, "Mobility report: On the pulse of the Networked Society," *White Paper*, 2016.

[2] Nokia Siemens Networks, "Understanding Smartphone Behavior in the Network," *White Paper*, 2011.

[3] D. Soldani, S. K. Das, M. Hassan, J. A. Hassan, and G. D. Mandyam, "Traffic Management for Mobile Broadband Networks," *IEEE Communications Magazine*, vol. 49, no. 10, pp. 98–100, 2011.

[4] A. Banerjee, "Revolutionizing CEM with subscriber-centric network operations and QoE strategy," *White paper, Heavy Reading*, July 2014.

[5] 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.

[6] P. Reichl, B. Tuffin, and R. Schatz, "Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience," *Telecommunication Systems*, vol. 52, no. 2, pp. 587–600, 2013.

[7] ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning," Dec. 1998, p. 8.

[8] ITU-T Recommendation G.1070, "Opinion Model for Video-Telephony Applications," Apr. 2007.

[9] R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the Quality of Experience of HTTP Video Streaming," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2011, pp. 485–492.

[10] G. Gómez, J. Lorca, R. García, and Q. Pérez, "Towards a QoE-Driven Resource Control in LTE and LTE-A Networks," *Journal of Computer Networks and Communications*, 2013.

[11] F. Lozano, G. Gómez, M.-C. Aguayo-Torres, C. Cárdenas, A. Plaza, A. Garrido, J. Baños-Polglase, and J. Poncela, "Network Performance Testing System Integrating Models for Automatic Qoe Evaluation of Popular Services: Youtube and Facebook," *Wireless Personal Communications*, vol. 81, no. 4, pp. 1377–1397, 2015.

[12] K. De Moor, I. Ketyko, W. Joseph, T. Deryckere, L. De Marez, L. Martens, and G. Verleye, "Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting," *Mobile Networks and Applications*, vol. 15, no. 3, pp. 378–391, 2010.

[13] K. Mitra, A. Zaslavsky, and C. Åhlund, "Context-aware QoE modelling, measurement, and prediction in mobile computing systems," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 920–936, 2015.

[14] 3GPP TS 23.203, "Technical Specification Group Services and System Aspects; Policy and charging control architecture, Version 13.1.0," Sep. 2014.

[15] H. Holma and A. Toskala, *LTE for UMTS-OFDMA and SC-FDMA Based Radio Access*. John Wiley & Sons (UK), 2009.

[16] K. I. Pedersen et al., "An Overview of Downlink Radio Resource Management for UTRAN Long-Term Evolution," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 86–93, 2009.

[17] F. R. M. Lima, T. F. Maciel, W. C. Freitas, and F. R. P. Cavalcanti, "Resource Assignment for Rate Maximization with QoS Guarantees in Multiservice Wireless Systems," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1318–1332, 2012.

[18] J.-H. Rhee, J. M. Holtzman, and D.-K. Kim, "Scheduling of Real/Non-real Time Services: Adaptive EXP/PF Algorithm," in *IEEE 57th Semiannual Vehicular Technology Conference (VTC 2003-Spring)*, vol. 1, 2003, pp. 462–466.

[19] X. Li, Y. Zaki, Y. Dong, N. Zahariev, and C. Goerg, "SON Potential for LTE Downlink MAC Scheduler," in *IEEE 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*, 2013, pp. 1–7.

[20] T. Farkhondeh, Y. Chan, and J. Lee, "Scheduling with Quality of Service Support in Wireless System," Jun. 25th 2014, EP Patent 2,277,329. [Online]. Available: http://www.google.com/patents/EP2277329B1?cl=en. Access date: October 2014

[21] S. Khan, S. Duhovnikov, E. Steinbach, and W. Kellerer, "MOS-Based Multiuser Multiapplication Cross-Layer Optimization for Mobile Multimedia Communication," *Advances in Multimedia*, 2007.

[22] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, P. Mogensen, and J. M. Lopez-Soler, "QoE oriented cross-layer design of a resource allocation algorithm in beyond 3G systems," *Computer Communications*, vol. 33, no. 5, pp. 571–582, 2010.

[23] P. Patras, A. Banchs, and P. Serrano, "A Control Theoretic Approach for Throughput Optimization in IEEE 802.11e EDCA WLANs," *Mobile Networks and Applications*, vol. 14, no. 6, pp. 697–708, 2009.

[24] W. He, K. Nahrstedt, and X. Liu, "End-to-End Delay Control of Multimedia Applications over Multihop Wireless Links," *ACM Transactions on Multimedia Computing, Communications and Applications (TOMCCAP)*, vol. 5, no. 2, 2008.

[25] H. Luo and M.-L. Shyu, "Quality of service provision in mobile multimedia - A survey," *Human-centric Computing and Information Sciences*, vol. 1, no. 1, pp. 1–15, 2011.

[26] P. Oliver-Balsalobre, M. Toril, S. Luna-Ramírez, and J. M. Ruiz Avilés, "Self-tuning of scheduling parameters for balancing the quality of experience among services in LTE," *EURASIP Journal on Wireless Communications and Networking*, no. 1, 2016.

[27] 3GPP TSG-RAN1#48, "LTE physical layer framework for performance verification, R1-070674," Feb. 2007.

[28] ITU-T Recommendation G.114, "One-way transmission time," May 2000.

[29] J. Navarro-Ortiz, J. M. Lopez-Soler, and G. Steay, "Quality of Experience Based Resource Sharing in IEEE 802.11e HCCA," in *IEEE European Wireless Conference (EW)*, 2010, pp. 454–461.

[30] P. Muñoz, I. de la Bandera, F. Ruiz, S. Luna-Ramírez, R. Barco, M. Toril, P. Lázaro, and J. Rodríguez, "Computationally-Efficient Design of a Dynamic System-Level LTE Simulator," *Int. Journal of Electronics and Telecommunications*, vol. 57, pp. 347–358, 2011.

[31] J. M. Ruiz-Avilés, S. Luna-Ramírez, M. Toril, and F. Ruiz, "Traffic steering by self-tuning controllers in enterprise lte femtocells," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 337, 2012.