Automatic best wireless network selection based on Key Performance Indicators

Stefano Boldrini, Maria-Gabriella Di Benedetto, Alessandro Tosti, and Jocelyn Fiorina

Abstract Introducing cognitive mechanisms at the application layer may lead to the possibility of an automatic selection of the wireless network that can guarantee best perceived experience by the final user. This chapter investigates this approach based on the concept of Quality of Experience (QoE), by introducing the use of application layer parameters, namely Key Performance Indicators (KPIs). KPIs are defined for different traffic types based on experimental data. A model for an application layer cognitive engine is presented, whose goal is to identify and select, based on KPIs, the best wireless network among available ones. An experimentation for the VoIP case, that foresees the use of the One-way end-to-end delay (OED) and the Mean Opinion Score (MOS) as KPIs is presented. This first implementation of the cognitive engine selects the network that, in that specific instant, offers the best QoE based on real captured data. To our knowledge, this is the first example of a cognitive engine that achieves best QoE in a context of heterogeneous wireless networks.

Stefano Boldrini
Department of Information Engineering, Electronics and Telecommunications (DIET), Sapienza University of Rome, Rome, Italy, and Department of Telecommunications, Supélec, Gif-sur-Yvette, France, e-mail: boldrini@newyork.ing.uniroma1.it

Maria-Gabriella Di Benedetto
Department of Information Engineering, Electronics and Telecommunications (DIET), Sapienza University of Rome, Rome, Italy, e-mail: gaby@acts.ing.uniroma1.it

Alessandro Tosti
Telecom Italia, Italy, e-mail: alessandro.tosti@telecomitalia.it

Jocelyn Fiorina
Department of Telecommunications, Supélec, Gif-sur-Yvette, France, e-mail: jocelyn.fiorina@supelec.fr

1 Introduction

Coexistence of different types of wireless networks is common experience. Widespread mobile devices use different technologies to communicate and exchange data. In most cases, when multiple networks are available, that may be based on either same or different technology, devices may choose the one to use and also possibly migrate from one network to a different one. This is for example the case when both cellular and one or more Wi-Fi networks are present.

Several investigations that defined algorithms for migration from a wireless network to another one of a different technology (the so-called *vertical handover* [1]) do exist. These "traditional" vertical handover algorithms are mainly based on physical or network layer parameters, or the combination of these two. In particular, Signal to Noise Ratio (SNR) and Received Signal Strength Indicator (RSSI) are the most studied and used parameters (even if usually linked to other network layer parameters) for the handover decision due to their simplicity [2]. This is "paid", however, with a lack of reliability in their real networks conditions representation.

Another important aspect is that, by considering lower layer parameters, the decision is taken with an eye on networks conditions; this is of course important, but only partial. In the process of network selection, more focus should be put, however, on final user experience, that can be better described and taken into account by the introduction of application layer parameters.

Moreover, network selection should be performed in an "intelligent" way, i.e. by adapting final decisions to a variety of factors such as the traffic type for which the connection needs to be established, networks current conditions and performance, as well as the used device performance.

This chapter aims at introducing the cognitive principle at the application layer by performing automatic best network selection based on "Key Performance Indicators". In other words, the final goal is the selection of the wireless network that can guarantee the best final user experience, thanks to the introduction of a *cognitive engine* that functions at the application layer.

To better understand and visualize this concept, a basic structure of the proposed model (deeply described in the following sections) is presented in Figure 1.

The chapter is organized as follows. In Section 2 the concept of "Quality of Experience" is introduced and it is explained how it can be obtained considering "Key Performance Indicators"; focus is put on the case of Voice over IP traffic type. Section 3 introduces the *cognitive engine*, whose behaviour and functionalities are described in Section 4. Section 5 presents an experimentation of the presented cognitive engine module in the Voice over IP case, while Section 6 contains conclusions and future work.

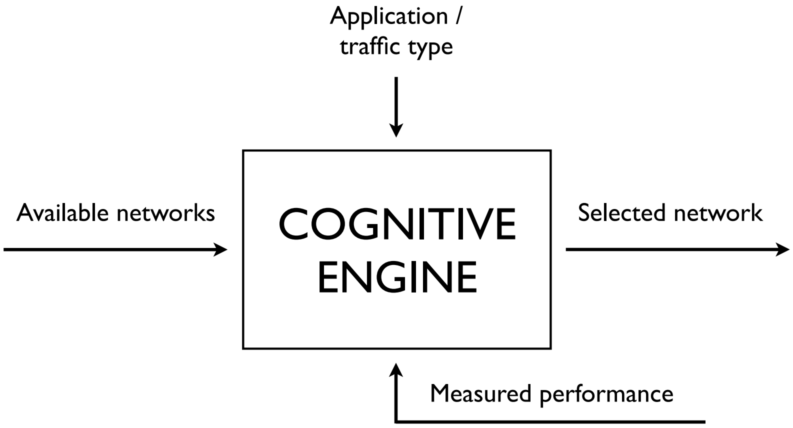

Fig. 1: Basic structure of the proposed cognitive engine.

2 Quality of Experience and KPIs

Quality of Service (QoS) is nowadays a fundamental aspect that Internet Service Providers (ISPs) have to take into account in order to offer different services with different guaranteed qualities at different prices. Parameters that are traditionally considered for QoS belong to physical layer (SNR/RSSI) or network layer (delay, jitter, throughput and packet loss). These values are the ones from which a QoS profile or QoS classification is built on. In other words, these parameters determine a classification in different traffic classes, each one with a different quality.

From the final user point of view, these parameters are only values that characterize its communication. What the user is really interested on, however, is the final quality perceived and experienced. This aspect is the reason for moving on from Quality of Service to "Quality of Experience" (QoE) [3], [4]. For example, the delay a network presents is an important factor that has impact on user's QoE; anyway, delay itself, considered alone and not in the whole context, does not completely reflect the quality the user effectively experiences.

Since the goal of offering a certain level of quality must focus on the final user, the quality that is effectively experienced must be pursued. There is the need of parameters that are better able to represent the perceived quality: "Key Performance Indicators" (KPIs).

KPIs are application layer parameters and therefore are much closer to the truly experienced quality. Given that these reside on a higher layer of the OSI model, they include and take into account the previous mentioned parameters, but in a wider and more comprehensive context. In fact they are able to consider all lower layers parameters and "synthesize" them by giving them the appropriate "weight" (for example in a linear combination, as presented in 3) based on the considered traffic type. SNR, delay, jitter, throughput and packet loss are therefore not important by themselves, but as part of more general parameters that incorporate them. Thanks to a learning process, KPIs are able to include also delays introduced by particular implementations of softwares and firmwares and specific behaviours of different devices using different telecom companies, aspects that are proved to be significant in the final user experience [5].

Until now, by our knowledge, application layer parameters have been introduced regarding minor aspects and in very specific cases [6], [7]. This chapter proposes to introduce an extensive use of these parameters for QoE evaluation.

KPIs can be defined for different traffic types, as for example voice communication, video and audio streaming, and web browsing. Each traffic type has its own peculiarities and "weaknesses", and therefore the attention on different aspects needs to be put based on the traffic type that is under consideration. For example, the delay a network presents is always important, but the impact it has on voice traffic type is considerably higher then in the case of web browsing traffic type; a similar thing can be said when considering jitter.

The identification and definition of the most suitable KPIs and their dependence on lower layers parameters for each traffic type can be done through an analysis of traffic data. Thanks to these data, the perceived quality can be correlated to the different layer parameters that result to be the most relevant (for the traffic type under consideration) and that therefore need to be considered for the KPIs definition. Traffic data used in this chapter for the definition of KPIs were provided by one of the major Italian telecom operator, that actively contributed in this work.

VoIP case

This chapter focuses on "Voice over Internet Protocol" (VoIP) traffic. This traffic type was specifically investigated because it represents nowadays an increasing relevance in Internet traffic (shown by the high popularity of specific software applications and the services offered by ISPs) and can also be an interesting study-case to test the proposed approach.

Two KPIs were identified to be relevant for VoIP traffic type:

1. One-way end-to-end delay (OED);
2. Mean Opinion Score (MOS).

OED, as the name says, is the unidirectional delay that is encountered from the sending node to the receiving node. Its value is the sum of every delay contribution introduced by each network node passed through. An indication of unidirectional delay values related to the quality of the communication can be found in [8]. International Telecommunication Union (ITU) indicates two threshold values: if the one-way delay is below 150 ms, the quality is very good; if the one-way delay is above 400 ms, the quality is very poor.

Based on this indication and on the provided data, in this chapter the following association between delay threshold values and perceived quality was used:

- if OED $\leq 150\,\text{ms}$, the communication perceived quality is very good;
- if $150\,\text{ms} <$ OED $\leq 250\,\text{ms}$, the communication perceived quality is quite good;
- if $250\,\text{ms} <$ OED $\leq 450\,\text{ms}$, the communication perceived quality is medium/poor;
- if OED $> 450\,\text{ms}$, the communication perceived quality is very poor.

MOS is a score that indicates the quality of a voice communication; it may vary in a range that goes from the minimum value of 1, that corresponds to a very poor quality, to the maximum value of 5, that corresponds to a very good quality [9]. It derives historically from the mean score assigned in tests with listeners in determined conditions. An association among MOS values, voice communication quality and perceived disturb can be found in Table 1.

Table 1: Association among MOS values, voice communication quality and perceived disturb.

MOS	Communication quality	Disturb description
5	very good	not perceivable
4	good	slightly perceivable
3	medium	perceivable but not annoying
2	poor	annoying
1	very poor	very annoying

In this chapter, two models for the MOS estimation were used. The first is described in [10] and can be expressed by the following equation:

$$\text{MOS} = 4 - 0.7 \cdot ln(\text{loss}) - 0.1 \cdot ln\left(\frac{\text{M} - \text{hsize}}{\text{drate}}\right) \quad ,$$

where “loss” is the packet loss expressed in percentage, “M” is the IP packet dimension expressed in bytes, “hsize” is the IP packet header dimension expressed in bytes, and “drate” is the used codec datarate expressed in kilobytes per second (kB/s). This model is valid in IP networks, and consider 4 as MOS maximum value. Other more complex models for MOS estimation can be found in [11], [12].

The second model used for MOS estimation derives from the provided traffic data and is summarized in Table 2. In this case, differently from the first model used, jitter is taken into account. A MOS value is assigned to a voice communication if it respects both the corresponding values imposed for packet loss and jitter (see Table 2). As an example, if from a determined number of sent packets it is obtained a packet loss of 2% and a mean jitter of 100 ms, a MOS value of 3 is assigned. Note that with this model only discrete MOS values are assigned, and that a MOS value of 5 is theoretically possible, even if practically quite impossible to obtain.

Table 2: Second model used for MOS estimation.

MOS	Packet loss (%)	Jitter (ms)
5	0	0
4	≤ 3	≤ 75
3	≤ 5	≤ 125
2	≤ 10	≤ 125
1	> 10	> 125

3 Cognitive engine

This chapter proposes the introduction of a module called “cognitive engine”, that can be implemented and installed in mobile devices. The final goal of the cognitive engine is to identify and select the wireless network, among the available ones, that permits to offer the best QoE for the final user. The network selection is based on KPIs, and for this reason is valid for a specific type of traffic,

Since the decision must be taken considering all the KPIs defined for the selected traffic type, a rule for the final selection that includes all of them must be defined. In this chapter, the definition of a *cost function* is proposed. In particular, a simple linear combination of KPIs is proposed as cost function. Given an application, i.e. a traffic type, a wireless network, and related KPI values, the cost is therefore expressed by the following equation:

$$c\left(\mathrm{KPI}_1,\ldots,\mathrm{KPI}_N\right) = \sum_{i=1}^{N} g_i\,\mathrm{KPI}_i \quad ,$$

where c is the final cost value of the network, N is the number of KPIs considered for the actual traffic type, and g_i is the gain for the i^{th} KPI.

It must be noted that each KPI presents a different gain g, i.e. has a different weight on the final decision. The gain values, that is to say how much a KPI is important for the cost within a specific traffic type, are determined by experimental data (together with the KPIs definition). However, the system presents a high flexibility. In fact the gain values can be updated and adjusted thanks to a learning process in order to refine the final selection based on the device specific behaviour (its firmware and software implementations, as better explained in Section 2).

Obviously, the goal is to obtain the lowest possible cost. This means that the selected wireless network is the one that presents the lowest cost. In this way, a soft decision is taken. However, for specific application or traffic types, it might be necessary to slightly modify this by introducing a hard decision rule. For example, in specific cases a KPI can be much more important than the others for QoE, and this can condition the final network selection.

4 Model structure

The cognitive engine is designed to be an intermediate layer of the system, considering an Open Systems Interconnection (OSI) protocol stack model. It is located right under the application layer, so that it can communicate directly with the applications that are running in the device. It is also in direct communication with the operating system (OS) of the device, in order to obtain information about the available wireless networks and the connection status of the current network in terms of lower layers parameters (SNR, delay, jitter, throughput, packet loss and every other parameter eventually necessary for the KPIs computation). This model structure is shown in Figure 2. Note that the cognitive engine is thought to be used for the wireless network selection, so all data not implied in the selection process can skip the transition through the cognitive engine and can directly pass from application to presentation layer.

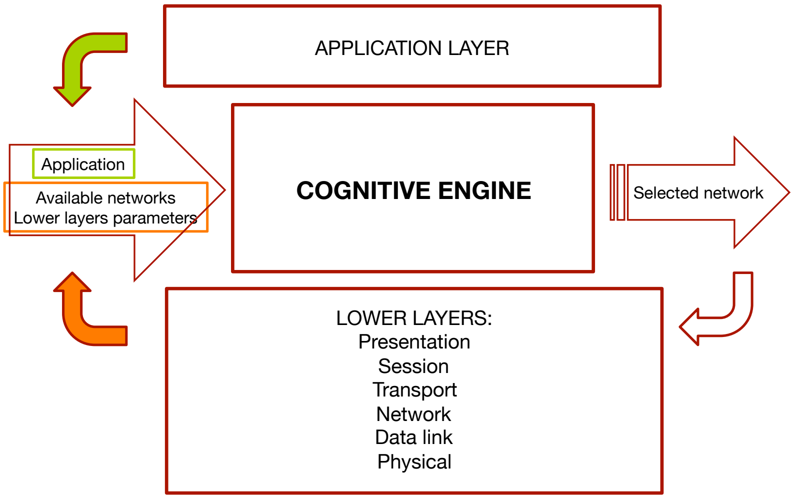

Fig. 2: The cognitive engine as intermediate layer and its location in the OSI system model.

The inputs of the cognitive engine are the following:

- the application that needs a connection (from the application layer);
- the available wireless networks (from the OS);
- lower layer parameters, eventually necessary for the KPIs computation (from the OS).

The output of the cognitive engine consists in the selected wireless network, that scores the best values of KPIs relevant for the running application (i.e. for the relative traffic type).

The functional behaviour of the cognitive engine can be outlined by the following logical steps:

- the application received as input is associated to one of the defined traffic types;
- once the traffic type is selected, the corresponding KPIs that need to be used and evaluated are identified;
- lower layers parameters that are needed for the KPIs evaluation are identified;
- for each available wireless network (whose list is received as input) lower layers parameters identified in the previous step are obtained:
  - from memory (input in the cognitive engine), if a previous measurement step was carried out;
  - by measuring; a "trial" connection is established if needed (this is the case, for example, of network layer parameters, that cannot be obtained otherwise);
- these parameters are used for KPIs evaluation, based on KPIs definitions and models; for each network there is, therefore, a set of KPI values;
- based on the KPI values, the cost function is computed for each network;
- the wireless network that presents the lowest cost is selected: it is the output of the cognitive engine.

Obviously, networks conditions may change: new wireless networks may be available, others may cease to be available (especially considering that the cognitive engine is thought to be implemented in mobile devices), and furthermore networks conditions may vary, so that the resulting KPIs may become significantly different from the values previously considered. Given this context, a periodical update must be performed in order to guarantee the choice of the best network under variable conditions. For this reason, the cognitive engine periodically updates the list of available networks and corresponding KPIs, by periodically repeating the steps presented above. This means that measures with the currently selected network are periodically performed and KPIs values updated; "trial" connections are again established for the other networks in order to have also their KPIs to be compared to the other values and, if convenient, a different network selection can be done. The frequency of this periodical update must be discussed separately since it involves many different aspects.

Moreover, the cognitive engine must incorporate a learning mechanism. Since each different device behaviour may introduce different delays and performance modifications that can significantly affect QoE [5], one of the cognitive engine task must be to "learn" from the device behaviour, to adapt to it, and to react as a consequence. To react means to consider the performance of the devices, i.e. to include, for example, the delay introduced by the specific implementation of the Internet browser or the VoIP application in the device where the cognitive engine is running. These behaviours cannot be known a priori, and for this reason a learning step is required.

Also the different gains for the different KPIs used for the cost function evaluation can be updated, adapted and modified. As a drawback, this learning phase can take some time, but it can be done in background as a "refining" process for the network selection. But the big advantage is a very refined selection method that completely takes into account *all* the aspects involved in the quality perceived by the final user.

A final consideration must be done on the so called “ping-pong effect”. After an update, if the selected network is different from the one selected in a previous stage, the device OS must connect to the new network in order to perform the best QoE. However, given the variability of the channels corresponding to the different networks (and especially if the cost variation is quite low), a same previous network could be selected in a following update. If the network change was performed, then at a next stage another change will be necessary, causing continuous and unnecessary network choice fluctuations, with the consequence of a waste of resources in terms of time and energy consumption spent for performing the changes. For this reason, in order to avoid this “ping-pong effect”, a latency on the decision or a hysteresis with threshold must be applied before effectively deciding for a network change. This also means that a series of frequent updates must be performed before proceeding with a change in the selected network.

5 Experimentation

5.1 Experimental set-up

A first experimentation was carried out considering the VoIP case. The parameters needed for the computation of OED and MOS (the two KPIs identified for VoIP traffic type, as presented in Section 2) are therefore the following:

- end-to-end delay;
- packet loss;
- IP packet dimension and its header dimension;
- used codec datarate;
- jitter.

Three of the above parameters (delay, packet loss and jitter) were obtained thanks to the use of the *ping* utility; the remaining (packet and header dimensions and datarate) were set as *ping* or KPIs inputs.

Packets sent with *ping* were sent from a computer towards a website server; this was chosen in order to always guarantee a minimum number of hops passed through and have therefore a realistic situation where the two end devices are connected through a certain number of intermediate nodes. In this case, there were always at least 15 hops from every location where the *ping* capture was carried out to the website server. For each capture, 50 packets of 64 bytes (dimension of IP packets) were sent. The obtained values are the result of the average of the 50 packets sent (and received back). Captures were taken at different times of the day during 10 days.

5 different wireless networks were used for the captures: 3 Wi-Fi networks and 2 different connections to the cellular network. These networks are located in different places in Rome, Italy; in every place, though, the minimum number of hops was respected. Although they are not present in the same place, they are an example of different wireless networks that can be effectively found in a same place and among which the device must choose. (They were chosen in different places due to captures bonds related to timing and capture device availability). The considered codec for VoIP communication is G.729 (CS-ACELP, conjugate-structure algebraic-code-excited linear prediction), that provides a datarate of 8 kb/s.

Values of the parameters obtained through these *ping* captures, together with the set values, were used to compute the KPIs for VoIP, and the final KPI values were then stored with the association to the time of the day when the capture was taken. In this first experimentation, a granularity of 15 minutes was considered, i.e. a capture was performed every 15 minutes during the central hours of the days.

5.2 Experimental data

OED values obtained are shown in Figure 3. For two of the networks (Wi-Fi network 3 and cellular network 2) data are available only for limited times of the day (between 14.15 and 15.15). Wi-Fi networks present in general lower delay (OED values are lower). Moreover cellular networks (in particular cellular network 1) show much more delay variability.

MOS values obtained are shown in Figure 4 (first model used) and in Figure 5 (second model used). It can be easily seen that the second model presents only discrete values. For both models the maximum value is limited to 4. It must be noted that when jitter is considered for MOS evaluation, i.e. in the second model used, cellular network 1 shows much lower MOS values in moments of the day when there is more variability in the delay.

In a first implementation of the cognitive engine, used for testing the described system, memorized data were used in order to select the wireless network that offers the best QoE for VoIP traffic; all presented networks were therefore thought to be available in the same place. Collected KPI data were normalized and used for network selection. Gain values chosen are 0.7 for OED and 0.3 for MOS, that is to say that end-to-end delay is considered to weight 70% on the QoE in a VoIP communication, and MOS is considered to weight the remaining 30%. These gain values can be, as explained before, adjusted and updated. Repeating the selection process at different times of the day gave as a result different networks, according to the memorized KPI values.

Cognitive engine must update the KPI values of the available networks before making the selection, in order to have the estimate of the experienced quality the more realistic as possible. However, having a database with KPI memorized data of the network that were already “seen” in the past permits to have a first estimate in case the update process is not possible before the application requires a connection

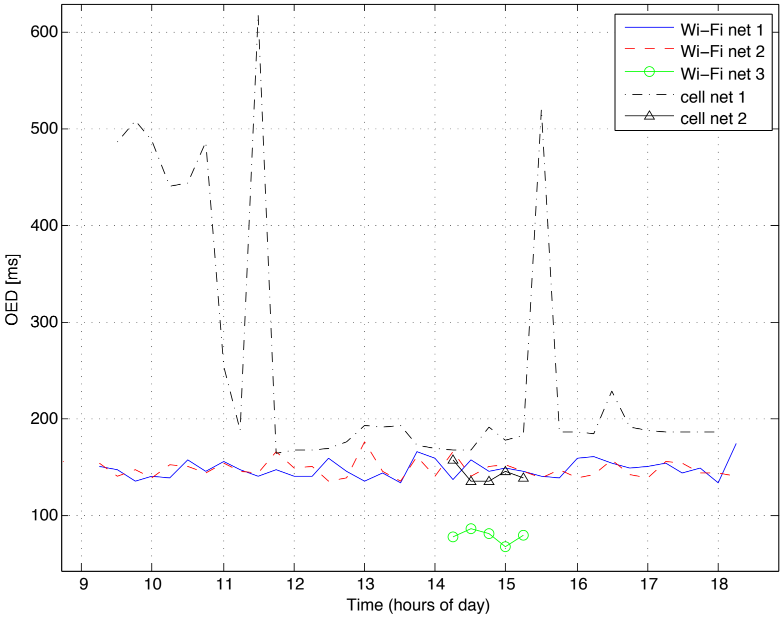

Fig. 3: OED values obtained at different times of the day with different wireless networks.

establishment (for example because there is no time to complete the update process before the connection starts). This estimate can be "rough" if it is based on few data, but it is at least a first basis on which the decision can be taken; moreover, as soon as possible, new values can be collected, data can be updated and the estimate can be therefore refined.

A consideration should also be done on the initial transitional period. In fact, when a new network, that was never "seen" before, becomes available, there is no stored data related to it. Until a new KPI update is performed, in order to have data also of this network, it is not selected even if it can present performance able to permit the best experience for the final user. In the period before the new update it is therefore present a transitory, where the best QoE is not fully guaranteed due to the lack of data.

6 Conclusions and future work

In this chapter, a cognitive architecture was introduced at the application layer: the wireless network that can guarantee the best experience to the final user was automatically selected thanks to the introduction and use of application layer param-

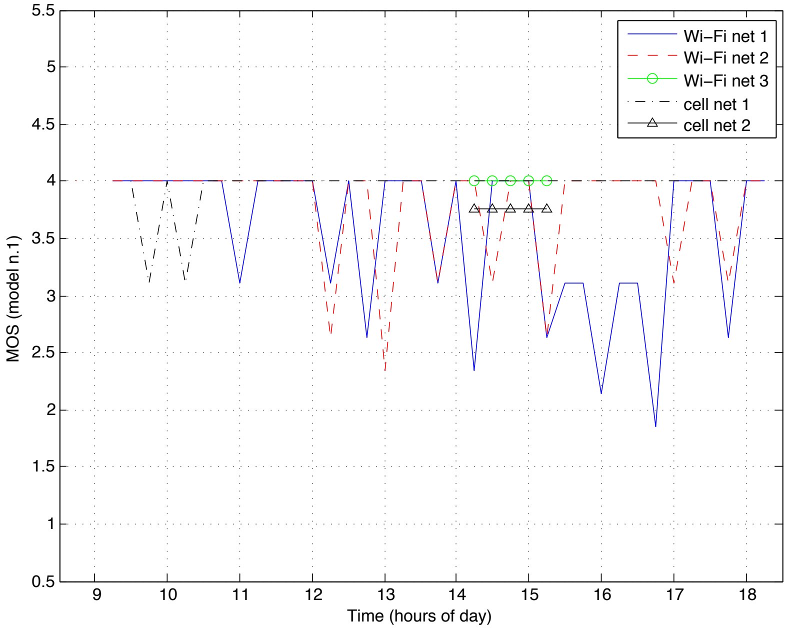

Fig. 4: MOS values obtained using the first model at different times of the day with different wireless networks.

eters, i.e. Key Performance Indicators. Quality of Experience was introduced and KPIs were defined for different traffic types based on experimental data. The model of a cognitive engine was presented, whose goal is to identify and select, based on KPIs, the best wireless network among the available ones. An experimentation was then carried out considering the VoIP case, with OED and MOS as KPIs.

From our knowledge, this is the first case in which application layer parameters are used in an extensive way, and the first example of cognitive engine with the goal of achievement the best QoE in a context of heterogeneous wireless networks.

The presented system presents high flexibility, since it can be applied in a general context, with different wireless technologies and with different types of traffic.

This cognitive engine model, that was tested in the VoIP case, should be tested with other traffic types, introducing the appropriate KPIs. Future work on this topic will also focus on the selection algorithm: the convergence time to the best network must be minimized, by taking into account the “multi-armed bandit problem”, i.e. how often the measuring (update) step should be performed and when avoiding wasting resources for the update process. Moreover, the presence of multiple users should be considered, since it may affect and modify performance of the different networks.

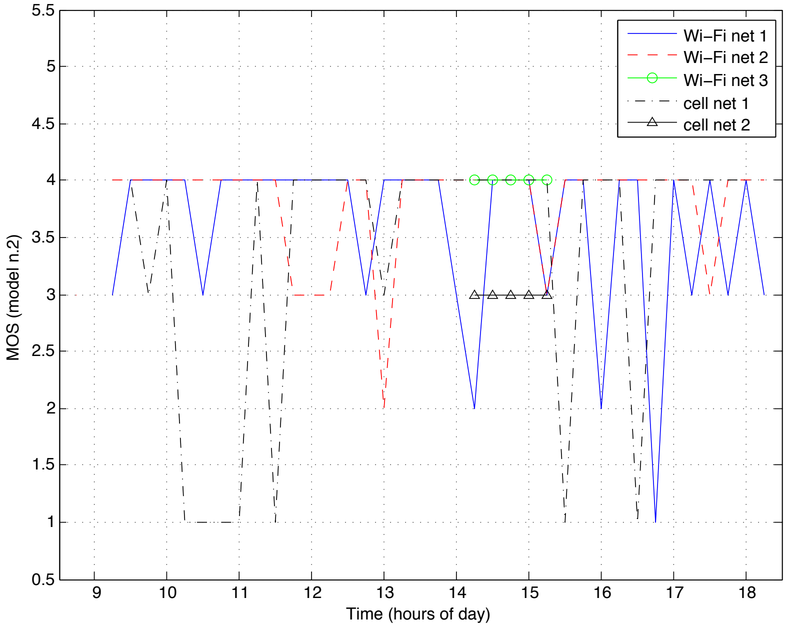

Fig. 5: MOS values obtained using the second model at different times of the day with different wireless networks.

Acknowledgements This work was partly supported by Telecom Italia, under contract between Sapienza University of Rome and Telecom Italia.

References

1. X. Yan, Y. A. Sekercioglu, and S. Narayanan, *A survey of vertical handover decision algorithms in Fourth Generation heterogeneous wireless networks*, Computer Networks, No. 54, 2010.
2. A. Ahmed, L. Boulahia, and D. Gaiti, *Enabling Vertical Handover Decisions in Heterogeneous Wireless Networks: A State-of-the-Art and A Classification*, IEEE Communications Surveys & Tutorials, Vol. PP, No. 99, 2013.
3. K. Piamrat, C. Viho, A. Ksentini, and J.-M. Bonnin, *Quality of Experience Measurements for Video Streaming over Wireless Networks*, 2009 Sixth International Conference on Information Technology: New Generations.
4. 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, 2012.
5. J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl, *Anatomizing Application Performance Differences on Smartphones*, MobiSys 2010.
6. C. Wang, T. Lin, and J-L. Chen, *A cross-layer adaptive algorithm for multimedia QoS fairness in WLAN environments using neural networks*, IET Communications, Vol. 1, No. 5, 2007.

7. P. Si, H. Ji, and F. R. Yu, *Optimal network selection in heterogeneous wireless multimedia networks*, Wireless Networks, Vol. 16, 2010, Springer.
8. ITU-T G.114.
9. ITU-T P.800.
10. L. A. R. Yamamoto, and J. G. Beerends, *Impact of network performance parameters on the end-to-end perceived speech quality*, Expert ATM Traffic Symposium 1997.
11. L. Ding, and R. A. Goubran, *Speech Quality Prediction in VoIP Using the Extended E-Model*, GLOBECOM 2003.
12. L. Sun, and E. C. Ifeachor, *Voice Quality Prediction Models and Their Application in VoIP Networks*, IEEE Transactions on Multimedia, Vol. 8, No. 4, 2006.