A SMARTPHONE AGENT FOR QOE EVALUATION AND USER CLASSIFICATION OVER MOBILE NETWORKS

Alessandro Verdolini and Stefano Petrangeli

TelecomItalia Laboratories- TILab, 10148 Turin, Italy
Polytechnic of Turin, 10129 Turin, Italy

ABSTRACT

The continuous growth of mobile users and bandwidth-consuming applications and the shortage of radio resources put a serious challenge on how to efficiently exploit existing networks and contemporary improve Quality of Experience. One of the most relevant problem for network operators is thus to find an explicit relationship between QoS and QoE, for the purpose of maximizing the latter while saving precious resources. In order to accomplish this challenging task, we present TeleAbarth, an innovative Android application entirely developed at TelecomItalia Laboratories, able to contemporary collect network measurements and end-users quality feedback regarding the use of smartphone applications. We deployed TeleAbarth in a field experimentation in order to study the relationship between QoS and QoE for video streaming applications, in terms of downstream bandwidth and video loading time. On the basis of the results obtained, we propose a technique to classify user behavior through his or her reliability, sensibility and fairness.

Index Terms— QoE, Android, Video Streaming, Mobile Network, User Classification

1. INTRODUCTION

In recent years, the relevance of Quality of Experience (QoE) has been widely studied and accepted by both Industrial and Academics actors [1]. The International Telecommunication Union (ITU-T) defines QoE as: *The overall acceptability of an application or service, as perceived subjectively by the end-user* [2]. With respect to Quality of Service (QoS), whose main purpose is to deal with technical network aspects, QoE reflects how the end-user perceives the service provided by the service provider. This aspect is crucial for operators in order to efficiently deploy, exploit and manage network resources. The emergence and growth of new real-time, bandwidth-consuming applications, not originally foreseen by network protocols [3], require operators to change their QoS-centric perspective to be QoE-centric. This shift leads to several advantages. Firstly, QoE allows operators to efficiently exploit current network resources. As pointed out in [4, 5, 6, 7], the link between QoS and QoE is far from linear; this means that an improvement of a relevant QoS parameter (bandwidth, delay, jitter, bit error rate etc.) does not entail a corresponding improvement, if any, in the user perception. Thus, it is more important for an operator to adjust QoS parameters in order to guarantee a certain QoE level to its users than merely improve them. Secondly, QoE is deeply tightened to users, i.e., it provides useful information that can be capitalized by operators to tailor network resources to the needs and behavior of a specific user [8].

In this article, we propose to investigate the above-mentioned QoE aspects, making use of an innovative mobile application developed at TelecomItalia Laboratories, which allows collecting network measurements and obtaining real-time explicit feedback from users about services or applications they use. Up to the authors' knowledge, this is one of the first attempts to study the correlation between QoS and QoE using measurements obtained from a real, operational mobile network and from users not gathered in a laboratory, thus entailing a great significance of the achieved results. In other works the communication channel was only simulated and users' feedback was collected inside laboratories with predetermined trials [5, 7] or was computed automatically [4]. Reference [6] used data obtained from a real fixed network, but it was impossible to trace measures back to correspondent users. We use the measurements collected by this application to develop an active QoS-QoE model (i.e. using explicit user feedback) and propose a mechanism to classify user behavior exploiting the above-mentioned model. It is worth stressing that this last point can be very important from an operator point of view, as it allows understanding the behavior of a particular user. Consequently, operators can adapt network resources in order to maximize his or her specific QoE, thus saving important resources. [8] already dealt with the problem of user classification, but they used a passive QoE model (i.e. obtained by just QoS parameters) as reference. Differently, in this article we adopt both an active QoE model and the feedback provided by each user, which gives useful information about the user itself.

The remainder of this article is structured as follows. Section II illustrates in detail the innovative platform developed to obtain QoS measures and users' feedback in order to evaluate QoE. Section III discusses the results obtained on field experimentation, for video streaming applications. Section IV introduces the mechanism used to classify users' behavior utilizing the results obtained in Section III. Finally, Section V presents a brief summary and proposals for future works.

2. DATA COLLECTION: TELEABARTH

Correlating subjective opinions, related with QoE, contemporary with actual network parameters (QoS) could be a very difficult task. The solution proposed in this article is TeleAbarth (TA), an innovative agent entirely developed at TelecomItalia Laboratories that can be installed on Android Smartphones from 2.2 release on. TA mainly aims at evaluating the performance of mobile services in terms of QoE with an end-to-end approach. It works in a transparent way, without interfering with the normal User

Experience and with low resource consumptions in terms of battery and CPU. TeleAbarth gathers network measurements only during mobile data session connectivity while in Wi-Fi connection it switches to a hibernated state in order to save resources. TeleAbarth works with an end-to-end approach in order to facilitate benchmark among operators, devices, locations.

It is fundamental to stress that TeleAbarth evaluates QoS parameters analyzing the real traffic generated by the smartphone, without generating artificial traffic. This approach provides two strong benefits: (i) avoiding the consumption of traffic bundle and (ii) detecting the real characteristics of the user's traffic.

In detail, TeleAbarth provides measurements of a large number of parameters concerning: (i) coordinates of the measurement; (ii) minimum and average RSSI signal strength; (iii) mobile network related parameters (cell identity, access technology typology, mobile operator, APN and IP address, number of cell reselections related to its access technology); (iv) smartphone model and its version release; (v) throughput, volume and activity time measurements of monitored applications as well as the overall total traffic in average, peak and standard deviation both in upstream and downstream; (vi) established, failed and dropped data sessions and their access time, i.e. the lapse of time to establish a new data session to obtain a new IP address; (vii) number of voice calls and their duration; (viii) asynchronous events like holes in the radio electric coverage, dropping of data session, timeout in establishment of data session, etc. related to their geographic location and exact time of the event.

Moreover, TeleAbarth implements innovative algorithms to elaborate traffic characteristics without violating users' privacy so that, for example, it is possible to elaborate the characteristics of internet browsing traffic but not to know which particular *url* has been visited. In particular, algorithms based on fuzzy methodology have been deployed to analyze: (i) browsing traffic (number of visited pages, average and peak size of visited web pages, average and peak duration time to download web pages, inter-arrival time between two consecutive web pages); (ii) video characteristics in both streaming and progressive download methods (number of video streaming seen, average and peak duration, average and peak throughput in downstream during fruition of video streaming, average and peak video loading time).

Furthermore, TeleAbarth is able to push brief questions (so called *instant polls*) after a particular event is fired (such as the fruition of a YouTube video) or when needed. These *instant polls* appear in the notification top bar of users' smartphone and require users to answer a brief question regarding the perceived quality of the applications and services used. These instant polls need to be concise and immediate in order to collect a fast familiar perception of the user experience.

In light of the above, it appears clearly that TeleAbarth works with and integrates two different planes: the plane of QoS, through objective network measurements, and the plane of QoE, through the instant polls regarding user perception of the enjoyed services. A system of web servers has the task to collect objective and subjective measurements and to provide their aggregation, elaboration, statistical analysis and finally representation. Appropriate *web-server rules* elaborate the gathered data, in order to hide the complexity of the internal database and provide a groundbreaking way to correlate different key performance indicators.

A fundamental characteristic of TeleAbarth is to be very flexible. It is possible to monitor just needed services or applications and to manage the fleet of controlled Smartphones in order to collect different measurements, for example regarding different field experimentations. This task is accomplished by the web servers that manipulate smartphone's behavior such as: the timing on which data are collected and sent to the server, which instant poll are sent, which services are monitored and so on.

Since November 23 2011, in 58 operative weeks, TeleAbarth collected about 300 thousands measure records on an average of 20 different device models and 65 different smartphones. Forty-eight thousands operating hours have been monitored. About 20 different releases of the agent have been developed with semi-automatic installation. Currently TeleAbarth is not yet available on Google Play Android Market.

On these prerequisites, we engaged a new trial involving 31 users, equipped with different models of last generation Android smartphones and with a traffic bundle belonging to different mobile operators. The task of those users was to use their smartphones in everyday situations on different environment (at home, in mobility, at office). Besides, it was asked them to see every day at least two or more video streaming on YouTube and CuboVision® (the TelecomItalia video platform) and answer to the instant polls sent by TeleAbarth servers some minutes after the fruition of a video streaming. These instant polls concerned: (i) possible problems users encountered during the video streaming, such as *video freezing*, *blockiness, sluggishness of video opening, non-optimal audio*; perceived video loading time: immediate (under 2 seconds), fast (between 3 and 4 seconds), moderate (between 5 and 10 seconds), slow (above 10 seconds); (iii) the Mean Opinion Score of quality in a scale between Bad (1) and Excellent (5).

At the end of the trial, the same users were asked to answer one-time questions regarding their own overall sensibility to the problems encountered during video streaming, in a scale of imperceptible, perceptible but not annoying, perceptible and annoying, extremely annoying.

Overall, in two weeks of trial, we collected about 400 video streaming measurements, correlated to about 1200 users' answers.

3. QOE MODELLIZATION

In this section, we illustrate the outcome of the experimentation introduced previously. We firstly analyze the relationship between downstream throughput-QoE and successively the relationship between video loading time-QoE.

3.1. Relationship Downstream Throughput-QoE

The first QoS parameter selected for our analysis is the downstream throughput experienced by users during a video streaming. This choice seems reasonable and in accordance with other studies [4, 5, 7], since the radio access network represents very often the bottleneck of the entire network. Our analysis focuses on the possible relationship between the downstream throughput and the video quality experienced by users.

Fig. 1 reports the average downstream throughput during video streaming for each class of the Mean Opinion Score, from 1 (Bad) to 5 (Excellent). It shows that, as expected, an increase in the downstream bandwidth entails an increase of the Mean Opinion Score provided by users. In addition, the statistical test of chi-square strongly confirms this outcome and shows a high statistical significance of the gathered measures. Nevertheless, it should be

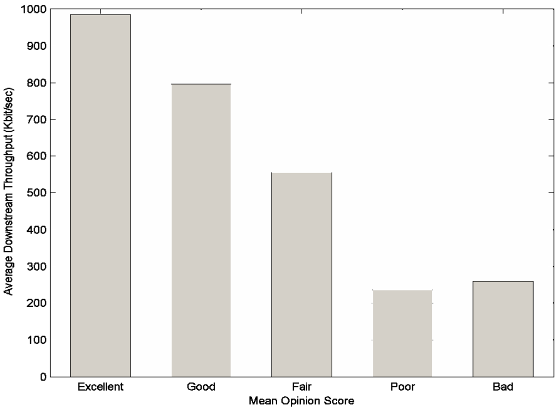

Figure 1: Mean opinion score versus average downstream throughput

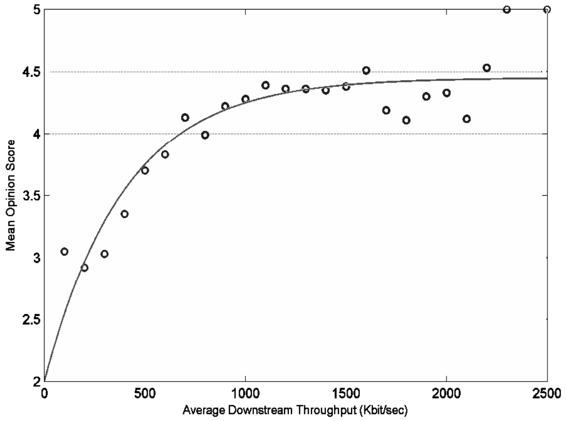

Figure 2: Mean opinion score versus downstream throughput

noted that not all the classes have the same statistical significance. The outcome shown for the class Bad can be explained considering that the answers gathered for this class are only a low percentage of the total.

The next step is to understand if a specific relationship exists between downstream throughput and QoE or not. Fig. 2 shows the result obtained from the experimentation object of this article. The x-axis of the graph reports the downstream throughput experienced during the video streaming. The y-axis reports the Mean Opinion Score of video quality. Each point of the graph is obtained averaging the answers associated to video streaming with similar downstream throughput. The graph starts from the point 2 (class Poor) instead of 1 (class Bad) since the average score for the totality of measures gathered is quite high and equal to 3.4. The relationship obtained is clearly exponential and a possible interpolating function is:

$$QoE_{videostreaming} = -2.45 \cdot 10^{-0.186 \cdot AverageThDownstream} + 4.45 \qquad (1)$$

where *AverageThDownstream* is expressed in bps. As already found in [4, 7] for VoIP applications and browsing, also our study seems to support the theory of an exponential relationship between QoE and downstream throughput. As it can be seen from Fig. 2, the saturation in the perceived quality is reached for a downstream throughput of 900-1000 Kbit/sec. This behavior could be explained considering that a good video codec for streaming applications is MPEG-4 that reaches a throughput of 1024 Kbit/sec considering the resolution of new smartphone screens. Above this threshold, any other increase in the bandwidth does not have any significant effect on the perceived quality. This aspect should be carefully considered by network operators in order to save precious resources. Future implementations of TA will implement this simple model in order to adjust the downstream throughput and take into account this result.

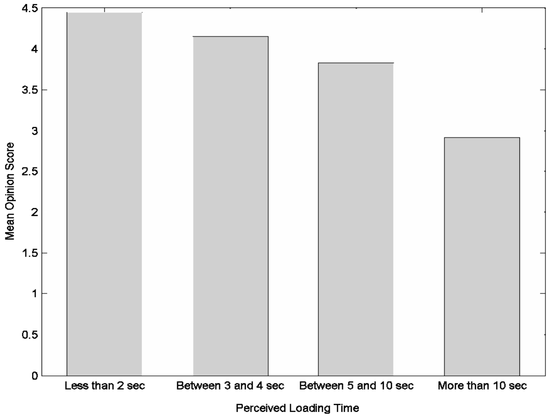

Figure 3: Perceived video loading time versus average Mean Opinion Score

3.2. Relationship Video Loading Time-QoE

Another possible variable affecting QoE we consider in this article is the video loading time. The question we would like to answer is if user perception of the video quality is influence or not by the immediateness/sluggishness of the video opening and possibly to what extent.

Firstly, we are going to analyze if perceived video quality is affected by video loading time. Fig. 3 reports the average Mean Opinion Score computed for each answer regarding the video loading time as perceived by users. It clearly shows that the perceived video quality is inversely proportional to the perceived video loading time. In particular, it can be seen that users consider annoying a perceived loading time above 10 seconds, in agreement with [9]. This result is supported analyzing the answers collected during the experimentation regarding problems experienced by users during a video streaming. When the loading time is considered a relevant impairment affecting video quality (38% of the total), the perceived loading time falls in the class “Above 10 seconds” in 67% of cases. It is fundamental to stress that the video loading time is not exclusively related to the available downstream bandwidth, but depends also on the playout buffering, network conditions, the presence of a content delivery network infrastructure. Fig. 4 confirms this statement since there is not a significant difference among the average downstream throughput for perceived loading time under 10 seconds.

As done for the downstream throughput, we also tried to evaluate quantitatively the influence of video loading time on the perceived quality. For this purpose, we have examined the loading time values computed by TA. In order to consider only relevant measures in our analysis, we have crosschecked video loading times computed by TA with users’ answers on this aspect and considered compatible ones. Fig. 5 and Fig. 6 show the outcome of our investigation. The x-axis of the graphs in Fig. 5 and Fig. 6 reports the video loading time computed by TA. The y-axis reports the Mean Opinion Score of video quality. Each point of the graph is obtained averaging the answers associated to video streaming

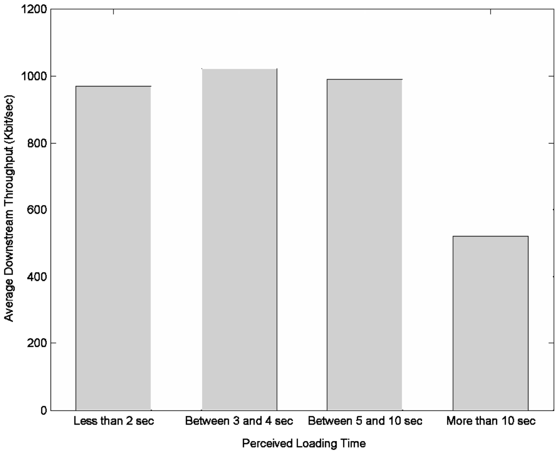

Figure 4: Perceived video loading time versus average downstream throughput

with similar loading time.

Fig. 5 shows a linear relationship between QoE and video loading time, and -2.2·*LoadingTime* + 4.5 as possible interpolating function. It is worth noting that the connection between video loading time and QoE is still present but weaker than the connection between downstream throughput and QoE. This evidence can be explained considering that the immediateness/sluggishness of the video opening is only an initial stimulus that can be weakened by the following fruition of the video. The result shown in Fig. 6 seems to support this suggestion. In this case we evaluate the influence of video loading time on QoE analyzing separately video streaming with a duration above and under 90 seconds. It is noticeable that long loading times affect more the perceived quality of video streaming with a short duration. It is also interesting to note that short loading times are rewarded in terms of QoE in short video streaming. As mentioned before, this is due to the fact that for short video streaming the initial stimulus (the video loading time) is less weakened than for long ones, and hence it provides a greater influence on the overall perceived quality.

4. USER CLASSIFICATION

This section deals with the problem of user classification, i.e. the establishment of a method able to associate to each user some macro-characteristics that can be representative of his or her behavior while using a specific application. The application object of the following analysis is video streaming, but it is fundamental to stress that the proposed approach is very general and can be applied also to other applications (e.g. browsing, VoIP, file download etc.) For the purpose of classification we exploit two type of information: (i) a local information relative to the behavior of a specific users, i.e. the answers he or she gave during the experimentation; (ii) a global information obtained analyzing the totality of answers gathered during the experimentation and the QoS-QoE model illustrated in Section III.

In the authors' vision, this type of analysis will become more and more important from a network operator point of view. Understanding and classifying the behavior of a user represents the first step to tailor, and possibly save, network resources in order to satisfy his or her QoE expectation. For example, if we were able to recognize that the perceived quality reported by a user is generally higher than expected or stipulated, we could re-allocate exceeding network resources to other users, without significantly worsening worsening his or her QoE. This approach would lead to a general increase of the quality experienced by users without enhancing network infrastructures, thus reducing CAPEX.

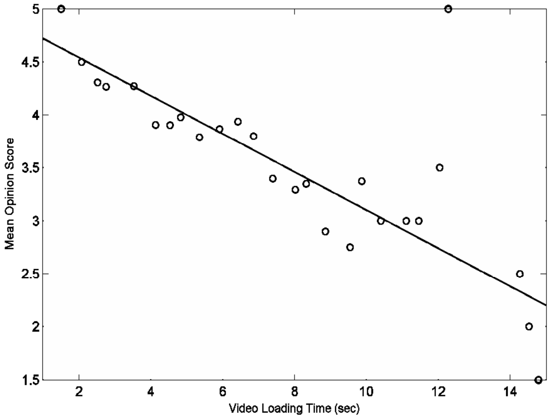

Figure 5: Mean opinion score versus video loading time

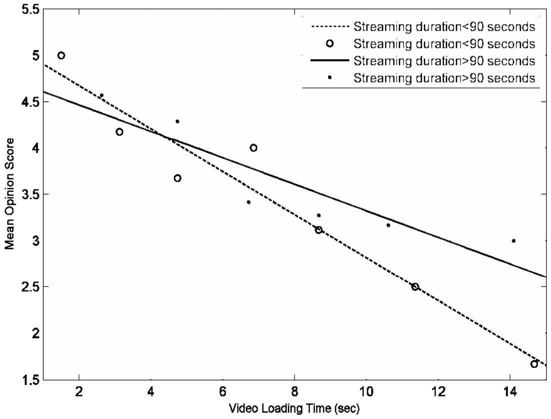

Figure 6: Mean opinion score versus video loading time for different video streaming durations

In this article, we propose a user classification based on three variables: (i) *reliability*, i.e. how much the quality feedbacks provided by a user agree with a reference model; (ii) *sensibility*, i.e. how much user perception is affected by QoS impairments experienced during the use of an application; (iii) *fairness*, i.e. how much the quality feedback provided by a user is in general higher or lower than a quality level considered as reference.

It is worth noting that this analysis can be executed periodically on a central sever in order to sharpen the classification on the basis of new measurements obtained.

4.1. Reliability

The first aspect we consider in our analysis is to understand if the quality feedback provided by a user is consistent with a model considered as reference or is, in the worst case, given randomly. This information can be used to stimulate unreliable users to give a more consistent feedback or to train them to use properly the quality feedback system.

For this purpose, we adopt a modified chi-squared test. Let us consider a user and the set of his or her associated answers on the quality of video streaming seen. We then arrange a first matrix M_u with five rows and three columns. Each row i refers to the possible answers of the Mean Opinion Score; each column *j* refers to an interval in the average downstream throughput experienced during video streaming (from 0 to 500 Kbit/sec, from 500 Kbit/sec to 1000 Kbit/sec, more than 1000 Kbit/sec). The generic element *(i,j)* of the matrix represents the number of answers given by the selected user, for the class *i* of the Mean Opinion Score and for the interval *j* in the average downstream throughput. We then arrange another matrix M_{tot} with the same structure of the one described previously. In this case the generic element *(i,j)* represents the number of answers given by the entirety of the users, for the class *i* of the Mean Opinion Score and for the interval *j* in the average downstream throughput. In Section III, we propose an explicit relationship between Mean Opinion Score and downstream throughput and for this reason we consider M_{tot} as matrix of the theoretical frequencies of the chi-square test, i.e. the matrix representing the average behavior of the users participating to the experimentation. In order to make the elements in M_{tot} comparable with the elements in M_u, we operate the following transformation:

$$\mathrm{M_{tot}(i,j)}=\mathrm{M_{tot}(i,j)}\frac{\sum_{j'}\mathrm{M_u}\left(\mathrm{i,j'}\right)}{\sum_{j'}\mathrm{M_{tot}}\left(\mathrm{i,j'}\right)} \qquad \text{for each i,j} \qquad (2)$$

The chi-square H_0 hypothesis is the fitness of the behavior of the considered user, represented by M_u, to the reference behavior, represented by M_{tot}. The chi-square test applied on M_u and M_{tot} will result in a value denoting the reliability of the selected user. The closer to zero the value obtained is, the less reliable the user can be considered. For example, the Reliability test applied to three users will result in the following:

Table 1: Reliability test

User	*Chi-square Test*	*Outcome*
A	86%	Highly reliable
B	18%	Sufficiently reliable
C	9%	Poorly reliable

4.2 Sensibility

Another relevant parameter we consider for user classification is his or her QoS sensibility, measured through the influence of specific impairments encountered during a video streaming on QoE. The impairments considered are *frequent video freezes*, *blockiness*, *sluggishness of video opening*, *non-optimal audio*. In order to evaluate the effects of such parameters on the experienced video quality, we explicitly asked users to give a score on their influence, at the end of the experimentation. The range of possible answers goes from 1- Extremely annoying, 2- Perceptible and annoying, 3- Perceptible but not annoying and 4- Imperceptible.

In order to proceed with the analysis on the sensibility, we decided to cluster our users in three different groups. The first one includes *demanding* users, i.e. users more affected than the average by video streaming impairments; the second one *average* users and the third one *insensible* users, i.e. users less affected than the average by video streaming impairments. We represent each user as a vector *U* containing his or her answers regarding the selected impairments. We apply to this set of vectors the Matlab function KMEANS, an implementation of the K-means algorithm, which allows to cluster a group of objects in K different clusters. In this article, we choose K equals to three, in order to represent the behavior of the three groups introduced previously. It should be noted that the K-means algorithm does not guarantee to find the optimal solution to the problem. In order to compensate for this issue, it is possible to apply the algorithm more than once and choose the most suitable solution among those computed. In our analysis, we decide to choose the solution that provides the *average* cluster more representative of the actual average behavior of the users participating to the experimentation. The K-means algorithm applied to our set results in the following centroids of the three clusters:

$$\begin{aligned}
\mathrm{Ctr_{demanding}} &= [2.6667 \quad 2.2222 \quad 2.1111 \quad 2.0000]\\
\mathrm{Ctr_{average}} &= [2.5455 \quad 2.2727 \quad 3.0909 \quad 3.7273]\\
\mathrm{Ctr_{insensible}} &= [4.0000 \quad 3.5000 \quad 3.7500 \quad 3.6250]
\end{aligned}$$

Each centroid represents the average behavior of a user belonging to a particular cluster. It can be clearly seen that the clusterization obtained well represents the three user classes *demanding*, *average* and *insensible*. In particular, the average scores of the three centroids are 2.25 for the demanding cluster, 3.71 for the insensible cluster and 2.90 for the average cluster (against an average of the entirety set equal to 2.89). We then associate to each user a triple computed as in the following:

$$\mathrm{AF_i}=\frac{\mathrm{D_i}}{\sum_j \mathrm{D_j}} \qquad \text{for} \quad \mathrm{i}\in\{\text{demanding, average, insensible}\} \qquad (3)$$

where AF_i, called affinity factor for the cluster *i*, quantifies how much a particular user does not belong to the considered cluster *i,* while D_i represents the Euclidean distance of the user associated vector *U* from the centroids $Ctr_{demanding}$, $Ctr_{average}$, $Ctr_{insensible}$. The closer to zero AF_i is, the more the user behavior is polarized toward the cluster *i*. For example, we applied the Sensibility test to the following three users:

Table 2: Sensibility Test

User	*User Answer*	$AF_{demanding}$	$AF_{average}$	$AF_{insensible}$
A	2 2 2 2	0.0379	0.3088	0.6533
B	2 2 3 4	0.4774	0.0430	0.4796
C	4 3 4 4	0.6318	0.3314	0.0369

We can clearly see that User A is very polarized toward the demanding cluster (the affinity factor is only 0.0379). In fact, the answers of User A depict a user very sensible to the impairments object of our analysis (all the impairments are considered "Perceptible and annoying"). Similar considerations can be repeated for User B and User C that are, respectively, polarized toward the average and the insensible cluster.

4.3. Fairness

The fairness parameter evaluates how much the quality feedback provided by a user is in general higher or lower than a quality level considered as reference. We define a user *fair* if his or her associated quality feedback agrees with the reference model, *unfair* or *enthusiast* if, respectively, is lower or higher than indicated by the reference model. It is worth stressing that identifying this behavior can be economically relevant for network operators. Unfair users are the most damaging for network operators since they are never satisfied of the service they use, even though the allocated network resources are adequate. Otherwise, enthusiast users are the most profitable, since they experience a good quality even though network resources are scarce.

In light of the above, we consider as reference model the exponential law (1) illustrated in Section III on the relationship between Mean Opinion Score and downstream throughput during video streaming. The fairness index *FI* associated to a specific user is computed as in the following:

$$FI=1-\frac{\int_0^{MAX_B}\left(MAX_{MOS}\text{-}QoE_{user}(x)\right)dx}{\int_0^{MAX_B}(MAX_{MOS}\text{-}QoE_{videostreaming}(x))dx} \qquad (4)$$

QoE_{user} represents the average Mean Opinion Score provided by the user when video streaming downstream throughput is equal to x (as in Section III we averaged answers associated to video streaming with similar downstream throughput). $QoE_{videostreaming}$ represents the exponential function indicated by (1). MAX_{MOS} denotes the maximum score in the Mean Opinion Score scale (5-Excellent) while [0; MAX_B] denotes the interval of integration. The denominator in (4) represents the area enclosed between the upper limit MAX_{MOS} and the exponential function (1) (i.e. the "enthusiast" area, relative to above the average quality feedback). The numerator depicts the correspondent area, but considering the function QoE_{user}, representing the behavior of the user object of the fairness analysis. Since in this case we have only a sequence of points, we approximate the integral by the trapezoidal rule, considering each point as belonging to the function QoE_{user}. The resulting fairness index *FI* is a number included in the interval [1-$4 \cdot MAX_B$/REF; 1], where REF is the value assumed from the denominator in (4). In order to analyze more easily the results, we normalize negative values of *FI* with the lower bound. In light of the above, the closer to zero *FI* is, the more the user can be considered *fair*. The closer to +1 or -1 *FI* is, the more the user can be considered, respectively, *enthusiast* or *unfair*.

We applied the fairness analysis just presented to a user whose answers are often above the average, as depicted in Fig. 7. The associated *FI* obtained applying (4) is 0.14, thus well representing the behavior of a user between the class *fair* and *enthusiast*.

5. CONCLUSION AND FUTURE WORKS

In this article, we presented a deepened and transversal analysis on the relationship between QoS and QoE for video streaming applications. Firstly we described an innovative and powerful Android application, named TeleAbarth, able to collect network measurements and explicit user quality feedback about services and applications running on smartphones. TeleAbarth has the important characteristic to be: (i) *flexible*, it is possible to dynamically change the applications and services monitored, the instant polls questioned to the users, the application configuration parameters and all the parameters regarding the experimentations without modifying the application code; (ii) *transparent with respect to the normal user experience and easily implementable on real devices*, every Android Smartphones and Tablets can potentially install TeleAbarth and collect measurements; (iii) *distributed*, only a small amount of computation is required to the central server, thus reducing the overhead signaling and internet traffic consumption.

We then presented the results obtained from a field experimentation involving 31 users, on the perceived quality of video streaming applications. In particular, we focused on the relationship between QoE-downstream throughput and QoE-video loading time. In the first case, we obtained an exponential relationship that agrees with previous studies on the same subject. In the second case, we obtained a linear relationship and gave a plausible explanation to this model. A technique to classify user behavior has successively been presented, using the results of the experimentation. Three variables have been introduced, reliability, sensibility and fairness able to efficiently model the behavior of a specific user.

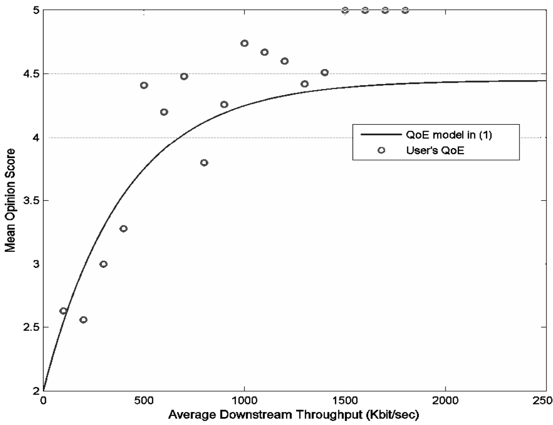

Figure 7: Fairness analysis

Future works will focus into three directions: (i) continuing the development of TeleAbarth in order to increase and improve its functionality; (ii) implementing the results reported in this article in the TeleAbarth system in order to close an ideal control loop directly to the user level; (iii) repeating the experimentation and the analysis presented for video streaming applications also for browsing, VoIP applications and online gaming, increasing the number of users.

6. REFERENCES

[1] K. Kilkki, "Quality of Experience in Communications Ecosystem", *Journal of Universal Computer Science*, Vol. 5, pp. 615-624, 2008.

[2] ITU-T Recommendation P.10/G.100, "New Appendix I – Definition of Quality of Experience (QoE)", 2006.

[3] B. Augustin, and A. Mellouk, "On Traffic Patterns of HTTP Applications", *IEEE Global Telecommunications Conference*, pp. 1-6, 2011.

[4] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A Generic Quantitative Relationship between Quality of Experience and Quality of Service", *IEEE Network,* pp. 36-41, 2010.

[5] K. Piamrat, C. Viho, A. Ksentini, and J.M. Bonnin, "Quality of Experience Measurements for Video Streaming over Wireless Networks", IEEE Computer Society, *Sixth International Conference on Information technology: New Generation*, Las Vegas, NV, pp. 1184-1189, 2009.

[6] S. Khirman, and P. Henriksen, "Relationship between Quality-of-Service and Quality-of-Experience for Public Internet Service", *3rd Passive & Active Measurement Workshop*, Fort Collins, 2002.

[7] P. Reichl, S. Egger, R. Schatz and A. D'Alconzo, "The Logarithmic Nature of QoE and the Role of the Weber-Fechner Law in QoE Assessment", *IEEE International Conference on Communications (ICC)*, Cape Town, 2010.

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

[9] N. Bhatti, A. Bouch and A. Kuchinsky, "Integrating User-Perceived Quality into Web Server Design", *Ninth International World Wide Web Conference*, Amsterdam, pp. 1-16, 2000.