From network-level measurements to expected Quality of Experience: the Skype use case

Salim Afra, Damien Saucez, Chadi Barakat

From network-level measurements to expected Quality of Experience: the Skype use case

Salim Afra, Damien Saucez, Chadi Barakat
Inria Sophia Antipolis – Méditérranée
Email: first.last@inria.fr

Abstract—**Applications rely on rich multimedia contents and experience of end users is sensitive to network conditions. Consequently, network operators must design their infrastructure to ensure high Quality of Experience (QoE) for their customers. However, applications are usually over-the-top services on which network operators have no control and users have no mean to tune the network when they undergo poor QoE. In this paper, we propose a method that allows network operators to determine how their network performance will influence QoE and end users to predict the QoE even before launching their applications. We predict the subjective QoE users will undergo based on the knowledge of objective network performance parameters obtained with active measurements (e.g., delay, loss) and machine learning. With the particular case of Skype calls and using a decision tree, we show that our approach achieves 83% of accuracy when estimating QoE from the delay, bandwidth, and loss. Our method can be seen as a new way of performing measurements at the Internet access, where instead of expressing the expected performance in terms of network-level measurements, the performance of the access is expressed in clear terms related to the expected quality for the main applications of interest to the end user. The strength of the approach is in its capacity of expressing directly the QoE as a function of network-level measurements, which is an enabler for QoE prediction, and in reusing the same network-level measurements as input to different models for the QoE of end user applications.**

I. Introduction

The Internet usage has changed drastically the recent years. We moved from an Internet meant to connect hosts together with basic offered services such as email, file transfer, or remote connection, to an Internet that is meant to consume digital content and that hosts plethora of services of all types including voice over IP (VoIP), video streaming, online shopping, games, news, etc. This has in particular changed the expectation end users have in terms of the quality of the Internet service. We moved from an era where the best effort nature of the Internet was driving the expectations of end users, mostly knowledgeable in Internet technology and Internet protocols, to an era where we want the best possible quality from the Internet and where any interruption of any service can cause frustration at end users with a serious economical impact. This important change has increased the pressure on network operators and service providers, more and more interested by capturing the Quality of Experience at end users, rather than simply capturing the physical properties of their Internet connection (e.g., bandwidth, delay, jitter, or loss rate). For example, a service provider cannot afford to wait for customers' complaints for having low quality of experience for their main applications, as according to a recent Accenture survey [1], about 90% of users simply change their network provider when they undergo low service quality without even taking time to give feedback to their operator before leaving. Therefore, it is essential that network operators and service providers have means to continually measure the *Quality of Experience* (QoE) and improve it as necessary. The difficulty being that, as compared to *Quality of Service* (QoS), QoE is a subjective measure of end users' satisfaction with the service they are getting from the network.

On the other hand, end users themselves are mostly equipped with measurement tools, [2], [3], that observe physical performance of their Internet connection (e.g., bandwidth, delay, and loss rate) without providing Quality of Experience information. Despite the presence of basic QoE indicators in popular applications like Skype [4] or Viber [5], there is a lack of a general solution for a fine-grained evaluation of the quality of Internet access in terms of QoE. A general solution to assess QoE would help end users and providers as on the one hand it would provide end users with a feedback that they can easily understand, and hence avoid confusion in the way they interpret the performance of their Internet access, and on the another hand, it would give providers an invaluable means for the profiling of the Internet accesses in terms of applications that can run with their QoE, hence allowing them a more transparent relation with their customers.

We propose a general solution for the estimation and prediction of the Quality of Experience for applications and services at the Internet access. The approach we follow consists of transforming measurements of performance at the network level as done today by most of the tools, into understandable terms at the user level function of the quality the users should expect for the applications and services of interest to them. For example, with our solution, an end user interested in a voice over IP application and/or a video streaming application should see indication on the expected quality for these two applications, rather than simple measurements of bandwidth and delay as is the current state of the art. This fine-grained feedback on the quality has to be done in a standalone way outside the applications themselves and even without the need to launch them, which is a required property for quality prediction and future communications planning.

The originality of our approach is twofold. First, we establish direct links between the subjective Quality of Experience (QoE) and the objective network-level measurements (QoS), mostly active measurements as bandwidth, delay, and loss rate. Machine learning techniques as decision trees can be used to capture such links by providing models for the QoE. Compared to the large literature on QoE, this direct linking with network-level measurements is novel and key property of our approach. The link is usually established with measurements done inside the application itself and on the data of the application (e.g., losses and delay experienced by the packets of the application), the case of Skype quality indicator being a typical example. Thanks to our approach, one does not need to run the application itself to estimate the expected quality, but rather estimate the quality with a standalone monitoring solution dedicated to this end. This direct linking to network-level measurements is key for the second novel property of our approach, the one of being able to reuse the same network-level measurements for the estimation of the quality of different applications. Indeed, thanks to QoE models function of network-level measurements, we are able to perform measurements once, and feed them to the different models in parallel, hence allowing scalability of the solution and its extension to as many applications as one can model.

In this paper we explain our approach with the typical Skype use case. We first start by explaining the principle of direct linking of QoE to network-level measurements and highlight the different techniques and experiments needed to this end. In particular, we explain how with lab experiments, we are able to change network conditions, note the expected Skype quality, and calibrate a statistical model based on decision tree for Skype quality estimation. We define the metrics of interest to Skype and implement a sampling method allowing to efficiently scan the large multi-dimensional space resulting from the joint consideration of these metrics. We give a snapshot of the decision tree that models Skype quality as a function of network performance and explain how it has been validated, both in terms of lab experiments and experiments in the wild using the PlanetLab platform, to capture at the end the exact quality predicted by Skype in 83% of the cases, while for most of the other cases a neighboring quality grade prediction is provided by the tree. Finally, we discuss the extension of the approach to further applications and operational issues related to the selection of the paths to measure.

II. From network-level measurements to QoE estimation: Our general approach

Our objective is to depart from network-level measurements, and be able to give indication on the expected quality of experience, for as many applications as one can model. Network-level measurements are performed once at the access of the Internet, mostly via active measurements launched from the device of the end user (mobile, desktop, etc.) and the same measurement results are reused as input for the various QoE predictors of the different modeled applications. Next, are the different steps for the realization of such QoE estimation. In the following section, we detail further the approach with the particular Skype use case.

Choice of performance metrics. The first step is to decide on the network performance metrics to measure and that are the most relevant to the application under consideration. The classical approach is to start from a large set of metrics, perform a sensitivity analysis on them, then keep the most relevant ones to the application. We seek network-level metrics that can be easily and accurately measured (e.g., delay, packet loss rate, available bandwidth, etc.).

Data collection. The next step is to gather data linking the considered performance metrics to the quality of experience for each considered application. Two main approaches can be followed. The first approach, as followed by Skype, is to collect the feedback of end users together with measurements of the performance metrics. This approach is widely known in the literature as *crowd sourcing*. The second approach, as the one we use in this paper, consists of controlled experiments, where the network configuration is artificially changed and the impact on the QoE is noted. The advantage of this second approach is in its feasibility for applications and tools that are not widely deployed, and also in its capacity to test unlikely but critical corner cases. It can form a first step for the calibration of a new tool, before relying on crowd sourcing for a collection of data in the wild and further refinement.

As for QoE measurement, the general approach is to ask directly the end user about its appreciation of the quality. This can be done for all kinds of applications as web browsing, shopping, multimedia applications, to cite but a few examples. Some applications have already done the work of modeling the QoE of end users and implemented the result in the form of quality meter embedded to the application as is the case of Skype, Viber and, some quality meter plugins for web browsing. Using such meters when available is another option, that eases our task and allows us to build upon previous work. We follow this second option later for the Skype use case.

QoE modeling. The objective of this step is to analyze the collected data and establish (or calibrate) from it a model linking the network performance metrics to the quality of experience for the considered application. Later, when the network performance is measured, the QoE can be predicted without the need to have the application running. This problem is known in the literature as a supervised machine learning problem [6], where the available data (network measurements) include information about the correct way to classify the data (associated QoE). The already classified data form both the training and validation sets.

The literature is full with techniques to establish QoE models. Departing from the results of recent studies on video streaming [7] and motivated by their appealing readability feature, we chose the decision tree learning technique which offers, according to [7], comprehensive rules with clear thresholds of network parameters for the determination of quality values. The over-fitting problem (i.e., having the tree very dependent of the collected data) is taken into consideration by the process of tree pruning which removes branches of the tree of low frequency and filters existent noise in the data. A tree can finally be coded in a software as a set of rules where a rule is the set of branches and nodes connecting the root of the three to one of the leafs. A rule is no other than a set of conditions on the different performance metrics to read this or that value for the QoE. We give later a snapshot of the Skype QoE decision tree and discuss the corresponding rules.

Model validation. The calibrated model for QoE has to be validated. The most straightforward validation is to label part of the collected data as validation set, and check the performance of the model on it. This is unfortunately not enough as running the model in the wild raises the question of the faithfulness of the controlled experiments. Differently speaking, one has to validate that the network configurations made during the controlled experimental phase represents well the network conditions in the wild, and that the calibrated model still performs well. Our approach is to stress the model for QoE over a variety of realistic Internet paths, before making it available. As we will show later, one efficient way to span a variety of realistic Internet paths is to use the PlanetLab platform [8] which provides hundreds of machines spread over the world and available for our experiments.

Sampling the performance metrics space. The size of the space of performance metrics increases exponentially with the number of these metrics. This poses the problem of the choice of values for these metrics during our experiments. On one side we want the space of metrics to be well covered, and on the other side we want to reduce the number of experiments. This corresponds to a sampling problem, and to handle it, we rely on one particular method, called the FAST method (Fourier Amplitude Sensitivity Analysis) [9] known for its efficiency in identifying the most relevant points in the space to explore, even at the corners of the space. We provide further details on this method later. Interestingly, FAST naturally provides sensitivity analysis of the model output (QoE) versus the different input metrics.

Measuring network access in the wild: When our models for QoE are used in real environments, we have to decide on the Internet paths to measure. These paths should be relevant to and representative of the modeled applications. In IP terminology, this can be translated into the IP address (or the host) to probe for the calculation of the performance metrics at the access. Measuring a nearby address provides an optimistic view of the network performance. On the other side, measuring a distant IP address can seriously underestimate the network performance that the user would really have. We advocate the principle of multiple measurement points (i.e., measure to different IP addresses), apply the QoE prediction on each of the corresponding paths, and present to the user the span of its QoE. This principle is implemented in our application ACQUA [10] and has proven its utility in capturing the performance at the access and the troubleshooting in case of anomalies in [11]. We can assist the end user further by only showing him a subset of paths that fit better his application, or by weighting the paths differently according to the application traffic pattern. In this paper we consider the measured path to be equal to the application path, and leave the general problem of finding the ideal set of paths to measure for future work.

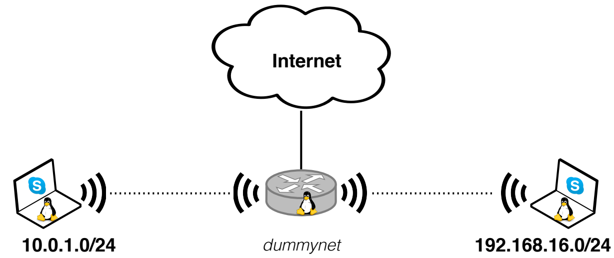

Fig. 1. Experimental testbed.

Next, we describe how this approach of different steps has been instantiated to the Skype use case.

III. The Skype use case

Quality of Experience is a subjective matter per se that requires the intervention of users. In the absence of annotated QoE dataset for Skype, we have to build our own dataset. Fortunately, in the latest versions of the Skype application, a quality indicator is provided during voice calls. It is therefore possible to construct a dataset that establishes the link between network performance metrics and Skype voice call QoE without requiring the intervention of a panel of users. To build the dataset we setup a fully controlled network environment and use smart sampling based on the FAST method to fairly span the multi-dimensional space relative to the considered network performance metrics. Part of the dataset is used as training set for our Skype Quality of Experience model while the remaining part serves as validation set. Sec. III-A details our measurement methodology. The validation set shows a prediction accuracy of the order of 80% (see Sec. III-B). In addition, we confront our QoE model trained with data obtained in the controlled environment with data we collect in a non-controlled network environment over PlanetLab and results show comparable accuracy levels (see Sec. III-C). This observation validates our assumption that Skype call quality can be well predicted with active measurements carried out at the network level outside the Skype application itself. It is worth to notice that the QoE indicator provided by the Skype application passively monitors the ongoing Skype traffic itself to determine the instantaneous QoE of a call, which makes it impractical for anticipative QoE prediction.

A. Dataset construction

To initiate a call, Skype needs to get connected to the Internet but once call established, packets are then sent directly between the caller and the callee without passing through a relay node thanks to the powerful NAT traversal mechanism included in Skype. This direct communication between hosts making a call permits to have a total control on the network conditions encountered during the call and hence the dataset.

As depicted by Fig. 1, our experimental testbed is composed of 2 hosts, playing the roles of caller and callee, connected together via a wireless network through a Linux access point. The role of the access point is twofold. On the one hand it provides an Internet access to the hosts[1] and on the other hand it emulates different network conditions. To force packets sent between hosts to pass via the access point, the two hosts are on separated IPv4 subnetworks and use the Linux access point as default IP gateway. Different network conditions in both directions are emulated using the Dummynet tool [12].

The Skype application being proprietary and requiring user actions, we have not been able to automatize the call procedure. To construct our dataset, we thus had to perform hundreds of manual calls. To that aim, we wrote a script that automatically sets up network conditions according to the FAST sampling method then waits for actions from the experimenter. When asked to do so, the experimenter launches a call at the caller node and picks up the call at the callee hosts then waits that the Skype quality meter stabilizes (it takes about 15 seconds). The observed meter value corresponds to the Skype QoE associated with the QoS metrics encountered during the call. The following 4 QoE values are possible:

- **Excellent**: Skype quality indicator is white;
- **Good**: Skype quality indicator is yellow;
- **Poor**: Skype quality indicator is red;
- **No Call**: the call cannot be established.

A key issue in our approach is the determination of the network performance metrics to measure and that have the most impact on the application QoE. The calibration of the QoE model within our lab will then consist of varying these performance metrics artificially and note their impact on the application quality. For Skype, and supported by recent findings in the literature on the quality of VoIP calls [13], [14], [15], we consider the following QoS metrics:

- Network end-to-end delay, both download and upload;
- Network packet loss rate, both upload and download;
- Network end-to-end available bandwidth, both download and upload.

This leads us to a total of 6 QoS metrics. An experiment will then consist of one instance of these six QoS metrics, that we use to configure the network via the Dummynet tool. We let Skype call run with this configuration until its meter converges and we note the corresponding quality. Then we move to another configuration, and so on. We limit the scope of the experimentations with one-way delays of up to 1000 ms, bandwidths of up to 1000 Kbps, and packet loss rates of up 50% that are known to be upper bounds for voice calls.

To minimize the number of experiments within this 6 dimensions space, we resort to sampling according to the smart *FAST* method (Fourier Amplitude Sensitivity Analysis) [9] as mentioned above. The idea of FAST is to introduce the notion of virtual time and assign to each parameter (i.e., QoS metric) a distinct integer frequency (characteristic frequency). By changing the virtual time, the different dimensions are jointly scanned. The number of experiments is set such that the spectrum of the QoE can be well captured. According to FAST, for a specific QoS parameter, the variance (or energy) contribution can be singled out of the model output with the help of the Fourier transformation. Therefore, FAST is also referred to as variance based sensitivity analysis. The advantage of using the FAST method is that it does not require any previous knowledge of the model to test. It is a method that samples the space smart enough to capture the impact on the QoE of possible changes in the input parameters. As given by the FAST method, we limit the experimentation space to 393 combinations of QoS parameters. In addition to these sampled combinations, we added 145 randomly chosen combinations for low loss rates. This set of 538 measurements constitutes our training set. We then performed 100 additional measurements randomly spread over the whole experimentation space to constitute a validation set independent of the training set.

[1]In order to establish Skype calls.

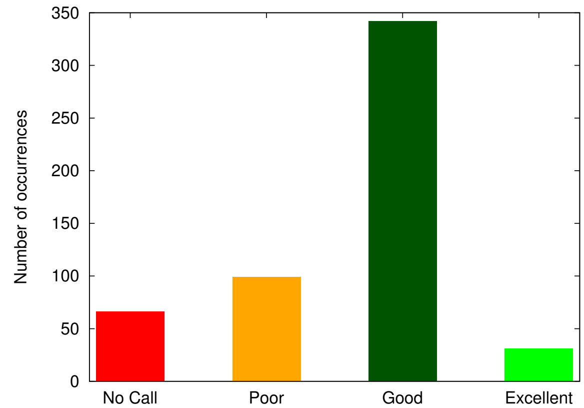

Fig. 2. Training set composition.

As long as all conditions are met (i.e., high bandwidth, low delay, low packet loss rate) call quality will be excellent. In a similar but opposite way, as soon as one network condition is disastrous (e.g., very high packet loss rate), no call is possible. On the contrary, the boundary between poor and good QoE is harder to discern as it is the result of the combination of all the QoS metrics. Consequently focusing more on the gray zone between a poor and a good quality of experience is the most efficient way to build the training set. Fig. 2 shows that our sampling strategy is adequate for our purpose as the two most represented QoE classes in the training set are the `Good` and the `Poor` ones with 63.6% and 18.4%, respectively.

B. *Prediction model and accuracy*

Our QoE prediction model relies on a C4.5 decision tree [6] built from the training set described in Sec. III-A. The raw decision tree is represented in Fig. 3. From that tree, one can see that the most decisive factor (i.e., the very first branching decision) is the loss rate in the upstream direction, where a loss rate higher than 27% definitely prevents a call to be of excellent quality, with more than 46%, calls are always poor or even impossible. This example, confirms that decision trees permit to readily understand the QoS conditions that have the major influence on the QoE, which is particularly useful for network architects and operators. In like manner, the tree

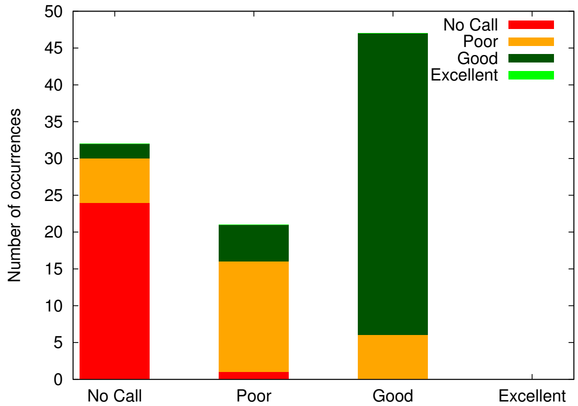

Fig. 4. Prediction accuracy breakdown.

shows that the effect of delay on the QoE is marginal for delays that are observed in practice.

To estimate the accuracy of the prediction model, there is two steps, first estimate the *tree construction accuracy* by itself, and then, the *prediction accuracy*. The first step is made by testing the learning set itself on the decision tree. A perfect tree construction would provide an accuracy of 100%, however, achieving such an accuracy would potentially incur a very large tree that would be subject to overfitting. The second step consists in testing a validation set independent of the training set. The tree construction accuracy in our case is of 85.7% which means that 85.7% of the observations in the learning set would be classified correctly. This accuracy must however be put in parallel with the fact that the tree is relatively large with 42 branches and hence rules and 99 nodes, implying the presence of numerous and complex decision rules. Also, to reduce the complexity of the prediction model and the risk of overfitting, C4.5 offers pruning that consists in removing branches, or part of branches, for which the construction accuracy is mediocre. The pruned tree we obtain is 26.3% smaller (73 nodes) than the initial tree but still keeps a high tree construction accuracy of 83.5% and can predict QoE of Skype calls from QoS measurements with only 20 rules.

The prediction accuracy is evaluated with the 100 observations contained in the validation set. The overall accuracy is 80%, which is of the same order as the tree construction accuracy. Fig. 4 details the prediction accuracy with the breakdown of prediction accuracy for each class of observation. Fig. 4 shows a propensity of the model to overestimate the quality of call in case of miss prediction but also a better accuracy for network conditions leading to good QoE. For instance, for the class of good QoE calls, 87.2% of predictions are correct while for lower QoE classes, the prediction accuracy falls below 75% with the lowest accuracy observed for the class grouping all poor quality calls (71.4%). As already stated this lowest accuracy comes from the intrinsically more complexity of identifying the exact boundary between conditions leading to a poor call quality of experience and conditions allowing to have good quality or no call at all. Nevertheless, it is important to notice that 90% of miss predictions are limited to a direct neighbor QoE class (e.g., predicted as good quality while the observation is indicating a poor quality).

C. *PlanetLab validation*

Our prediction model uses supervised machine learning with a C4.5 decision tree. To calibrate our model, we use a dataset built from the fully controlled environment as explained in Sec. III-A. With Sec. III-B we show that the model reaches a high level of accuracy. The usage of a fully controlled environment to construct the dataset and calibrate the model permits to precisely understand the impact of each network performance metric as they can be isolated from others. However, in practice Skype voice calls would be performed between devices interconnected through the Internet where cross traffic may impact network performances by influencing congestion and queuing variability. Therefore, to assess the accuracy of our model in real conditions, we evaluate it with measurements gathered in a non-controlled environment using PlanetLab [8]. The experimental setup used to produce this new dataset is the same as the one used in Sec. III-A except that instead of emulating network conditions using Dummynet, we enforce packets to be forwarded over arbitrary paths on the Internet. More precisely, to emulate calls between distant devices, we deflect packets, using address translation, from the local network to one of our relay nodes located in the Internet, on PlanetLab machines. As a result, packets exchanged during the call traverse Internet links with their inevitable cross traffic and queues. We selected 11 PlanetLab nodes, spread over all geographical regions, playing the role of relay for the calls. For each node, we performed 2 calls. The result of this experimentation show a prediction accuracy of 86.3% which is consistent with what we have observed with data gathered from the controlled environment. This indicates that measuring performance metrics that are only an approximation of the real network performance do not have any major impact on the prediction accuracy and that the effect of cross traffic is well captured by our QoS metrics.

IV. Conclusion

This paper proposes a new method to quantify the performance of Internet access. The aim is to allow network and service providers, and application users, to predict subjective QoE from the active measurement of objective network performance metrics such as delay or bandwidth. Controlled experiments and supervised machine learning are used to this end. The strength of our method is in its capacity to express the QoE directly as a function of network-level measurements that anyone can perform, thus allowing to predict the QoE of applications without the need to launch them. This method even allows to reuse the same set of network-level measurements to predict the QoE of different and independent applications, reducing so the cost of predicting QoE.

We demonstrate the practicality of our method with the particular case of Skype VoIP calls. Thanks to a decision

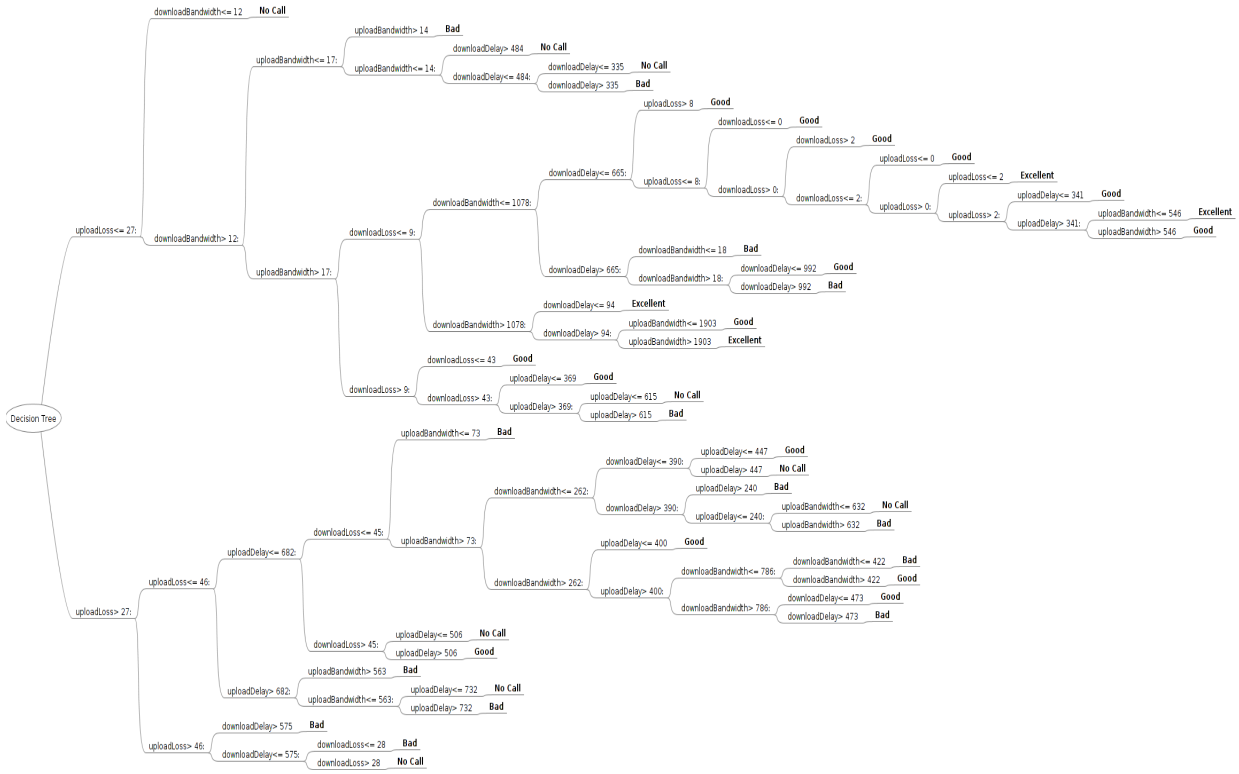

Fig. 3. Decision tree computed with C4.5 on the training set.

tree machine learning algorithm and extensive experiments in a controlled testbed but also in PlanetLab, we demonstrate that a QoE prediction accuracy of as much as 83% can be achieved by solely measuring delay, bandwidth, and loss rate key performance indexes.

As a future work, we are planing to validate and refine our Skype prediction model using crowdsourcing and construct QoE prediction models for other classes of applications such as video streaming.

REFERENCES

[1] Nokia, "Quality of Experience (QoE) of Mobile Services: Can It Be Measured and Improved?" Nokia, Tech. Rep., 2004.
[2] "CAIDA: Center for Applied Internet Data Analysis," 2014, http://www.caida.org/.
[3] "m-lab," 2014, http://www.measurementlab.net.
[4] "Skype, free calls to friends an family," 2014, http://www.skype.com.
[5] "Free calls, text and picture sharing with anyone, anywhere!" 2014, http://www.viber.com.
[6] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007, pp. 3–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=1566770.1566773
[7] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM '13. ACM, 2013.
[8] "Planetlab europe," 2014, www.planet-lab.eu.
[9] R. I. Cukier, C. M. Fortuin, K. E. Shuler, A. G. Petschek, and J. H. Schaibly, "Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory," *The Journal of Chemical Physics*, vol. 59, no. 8, pp. 3873–3878, 1973. [Online]. Available: http://link.aip.org/link/?JCP/59/3873/1
[10] "ACQUA: Application for Collaborative Estimation of QUality of Internet Access," 2014, http://planete.inria.fr/acqua/.
[11] R. Cascella and C. Barakat, "Estimating the access link quality by active measurements," in *Teletraffic Congress (ITC), 2010 22nd International*, Sept 2010, pp. 1–8.
[12] L. Rizzo, "Dummynet: A simple approach to the evaluation of network protocols," *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 1, pp. 31–41, Jan. 1997. [Online]. Available: http://doi.acm.org/10.1145/251007.251012
[13] M. S. Islam and S. N. Mehdi, "How different qos mechanisms affect voip qos metrics," p. 47, 2010.
[14] X. Gu, K. Nahrstedt, R. Chang, and Z.-Y. Shae, "An overlay based qos-aware voice-over-ip conferencing system," in *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, vol. 3, June 2004, pp. 2111–2114 Vol.3.
[15] L. Yamamoto and J. Beerends, "Impact of network performance parameters on the end-to-end perceived speech quality," in *In In Proceedings of EXPERT ATM Traffic Symposium*, 1997.