32nd IEEE Conference on Local Computer Networks

Measurement Based Call Quality Reporting

René Serral-Gracià, Loránd Jakab, Jordi Domingo-Pascual*
Universitat Politècnica de Catalunya
{rserral, ljakab, jordid}@ac.upc.edu

***Abstract*— Real-time traffic is an important issue when designing networks. The growth in the deployment of voice applications in packet switched networks requires that the network can guarantee a minimum level of quality. At the same time, customers want guaranties that the proper Quality of Service is provided for their contracted services. In this environment service providers need means of reporting the quality level of their networks.**

Standardisation bodies define general metrics, such as delay, jitter or losses without giving detailed information regarding the actual user's perception of the traffic. Other approaches such as the *Mean Opinion Score* (MOS), are too coarse to be significant on *best-effort* based environments such as the Internet.

This work is motivated by the potential change in the Quality of Service (QoS) parameters found in packet switched networks. Which highlights the inaccuracy of metrics such as MOS, since they were designed for the stable circuit switched networks.

This paper proposes a novel approach to on-line voice quality measurement which is more suitable for packet switched networks than the classical alternatives, while keeping the subjective approach of MOS. Moreover, our solution permits the real-time reporting of the voice communication's quality.

Along with the specification of this metric, the paper provides an experimental validation of the proposal using a real testbed with real applications. The results show the higher accuracy in reporting using this proposal over the standard metrics currently used.

I. Introduction

Currently in the Internet, voice applications use real-time traffic to transmit the conversation data. Usually such traffic is transmitted using protocols such as RTP or UDP, that might be considered as low priority in some environments. This lack of guaranties in the voice delivery makes it difficult for operators and companies to offer Quality of Service (QoS) for these services.

This paper proposes a metric to quantify the user's experience in voice communications over packet switched networks. We designed this metric using the proposals defined by ITU [12] and IPPM [1] groups as starting points. It permits a higher level of accuracy than the existing alternatives.

The most used standard metric for call quality reporting is the *Mean Opinion Score* (MOS). Initially, MOS was designed for classical telephony conversations. This trend is slowly changing as more and more companies are providing VoIP services (AT&T, Verizon, Skype, etc.) through the Internet. This unveils the need of a more verbose metric to parametrise user's perception of the communication in such a changing environment. Because under our point of view a single MOS value is not sufficient to summarise a whole conversation in these conditions.

On the other hand, network level metrics and measurements have no knowledge of higher layer information that is relevant to report the quality of the voice transmission, for example the codec, which determines the effect over the quality in the case the network conditions are not optimal.

In order to overcome these limitations this paper presents a new metric which combines the application layer information provided by MOS, and the accuracy reported by the network level metrics over time. This approach offers several advantages:

1) The operators can evaluate the voice quality over the dynamic behaviour of the network, and react accordingly in case it is needed.
2) Quickly detect and report deviations from contractual QoS level.
3) Predict the potential service failure in advance.

This work studies in detail the first advantage leaving the rest as an important part of our future work.

Besides the metric proposal, we also performed some tests in a real network to validate the metric. We used a well known VoIP application in a series of different network situations. The network is stressed in different ways from severe packet losses to high network delays in order to obtain a broad range of situations and deliver a proper set of results.

The testing and results have been performed using the expertise acquired in measurements in [5], [24] and also in the framework of the EuQoS project [9].

The rest of the paper is divided as follows. First we show the related work in the performance metric and VoIP fields. Later the paper focuses on the initial background of our work which permits to develop the metric definition in section IV. The validation and results are shown in section V, and finally section VI concludes and shows the work left for future study.

II. Related Work

The emerging companies and services using VoIP technology makes it an interesting topic of research. Our work on VoIP quality reporting is complementary to already existing solutions in this area. A. Raake in [2] discusses the effects of various network parameters on voice traffic.

Besides the pure quality assessment, often the studied aspects of VoIP are centred on its deployment in real testbeds [6], [8]. The spread of wireless technologies poses new challenges on VoIP deployment [21], [27]. Where many efforts are

*This work was partially funded by IST under contract 6FP-004503 (*IST-EuQoS*), MCyT (Spanish Ministry of Science and Technology) under contract TSI 2005-07520-C03-02 and the CIRIT (Catalan Research Council) under contract 2005 SGR 00481.

0742-1303/07 $25.00 © 2007 IEEE
DOI 10.1109/LCN.2007.76

IEEE computer society

Authorized licensed use limited to: UNIVERSITAT POLITÈCNICA DE CATALUNYA. Downloaded on May 13, 2009 at 11:49 from IEEE Xplore. Restrictions apply.

centered in the deployment cost, feasibility and performance. In [21] the authors analyse VoWF (VoIP over Wireless) with the goal of substituting cellular systems or, at least, reduce the costs of voice transmission by using 802.11 technology where available.

Codecs and their behaviour are also deeply studied in the literature [16], [17], [15], [20]. Specifically Jiang and Schulzrinne in [16] analyse the on-off patterns in VoIP, they focus on the silence detection algorithms found in some codecs. The same authors in [17] use speech recognition as a way of predicting the perceived quality, using MOS and the E-Model [13] to do that.

From the measurement point of view, MOS analysis in real networks is a subject of research. First efforts in MOS modeling are discussed in [7]. The outcome of this work has been used in [3] where the authors analyse the effects of link failures on VoIP performance. Our work takes the same quality parameters, but with the goal of improving the reporting by means of a new metric, validating it using a real voice application.

All the above work differs from ours in the fact that we develop a method for quality reporting, without neither modifying nor evaluating any underlying network or codec.

Also related in the quality metrics analysis, some work was done in [18] by Lakaniemi et al., they focus on the impact of packet losses in standard conditions on domestic and international links. This analysis does not consider extreme network conditions where MOS might not behave as expected.

Jiang et al. perform some QoS evaluation using measurements in [15], the main difference from our work is that the authors focus on the Mouth to Ear delay, which is not possible to study from the network point of view taken by our approach.

There are also standardisation bodies who also use MOS for reporting the transferred voice quality. Specifically, in RFC-3611 [25], the authors define some RTCP control extensions to the protocol by reporting VoIP MOS values. Instead, we are proposing a full metric, not an extension of an application dependent solution. Moreover, our proposal can report the information both on-line and off-line in an integrated environment which helps customers and operators.

III. Metrics Background

The main contribution of this paper is to define a new metric, which is based on existing ITU's definition of the *Mean Opinion Score* (MOS). This section introduces the current MOS specification, along with the underlying information which will help the development of such metric. It also discusses some important concepts that will limit or conditionate the results. The actual metric will be explained in the next section.

A. MOS

MOS is a value ranging between 1 and 4.5. It defines the overall subjective quality of any voice communication, 4.5 being the maximum and 1 the worst achievable degree of quality. MOS value can be obtained through the E-Model [13] which gives a deterministic computation of a subjective value. Its formula can be found in Equation 1.

$$\begin{aligned} MOS_{CQE} = &\ 1, && R = 0 \\ MOS_{CQE} = &\ 1 + 0.035R + R(R-60)\cdot \\ &\ \cdot(100-R)\cdot 7\cdot 10^{-6}, && 0 < R < 100 \\ MOS_{CQE} = &\ 4.5, && otherwise \end{aligned} \tag{1}$$

Where R is known as the transmission rating factor, computed by the equation 2. In this equation R_o stands for the signal-to-noise ratio, I_s is the simultaneous impairment factor. I_d refers to delay impairment factor, I_{e_eff} is the effective equipment impairment factor, and finally A holds the Advantage factor.

$$R = R_o - I_s - I_d - I_{e\_eff} + A \tag{2}$$

R ranges from 0 to 100 and its quality degrees are enumerated in table I.

TABLE I
POSSIBLE R AND MOS RANGES

R (lower limit)	MOS (lower limit)	User Satisfaction
90	4.34	Everybody satisfied
80	4.03	Satisfied
70	3.60	Some users not satisfied
60	3.10	Many users dissatisfied
50	2.58	Nearly all users not satisfied
0	1	Impossible to understand the conversation

For further details on this topic, the reader is referred to ITU's recommendations [14], where the proposed implementation discusses several different environments and its effects on the final quality. For the ease of simplicity, we take the default values [7] on all the parameters, except the ones derived from the network behaviour, such as one-way delay and packet losses.

The most relevant parameters for this study are I_d and I_{e_eff}. I_d represents the mouth to ear delay (one way delay in our scenario) and I_{e_eff} that is codec dependent and is computed from the packet losses. They bound the quality limits of the communication in the network.

B. Network metrics

MOS is designed to use information only available on application layer (i.e. codecs), but when analysing a network in general such information is not available or too difficult to acquire.

Hence, when studying the network from lower layers other metrics must be considered. IETF's IP Performance Metrics (IPPM) is a group focused on standardising the different possible network metrics. All their definitions start from a generic framework [22] which defines all the bases that will derive on several metrics. Opposed to ITU the metrics defined

Authorized licensed use limited to: UNIVERSITAT POLITÈCNICA DE CATALUNYA. Downloaded on May 13, 2009 at 11:49 from IEEE Xplore. Restrictions apply.

by IPPM are as decoupled from the applications as possible, defining parameters to assess the one way communication quality and reliability such as *One-Way Delay* [10], *Inter-Packet Delay Variation* (IPDV) [4] or *Packet Losses* [11]. All these metrics permit to determine the network's quality level, but neither define nor explain the user's perception of that quality.

Despite all of the above, IPPM's framework defines the basis to enhance and extend existing metrics. This is based on the introduction of Type-P packets. A Type-P packet is defined as a generic IP packet which, depending on the metric will be instantiated to actual network packets. As an example, when studying one-way delays on a testbed, a `Type-P-One-way-delay` is defined as a packet with a `Source` address, which is sent towards its `Destination` and which transmission takes `T` time units. So, all measured packets having a Source, a Destination and transmission time (which means that are not lost) are suitable to be selected for the metric.

Together with MOS's E-Model, this work uses the above framework to derive a new metric definition (Extended MOS) which combines the user's perception provided by MOS with the lower and more accurate information obtained from the network layer.

IV. Extended MOS

The MOS metric was initially designed to describe the overall quality of a call on a subjective scale, based on the assumption that the call is routed through a circuit switched network. Today, as VoIP calls are routed over the Internet, we believe that it is not sufficient to describe the call quality with a single value. The duration of a call can last from a few minutes to as much as few hours. During this period the underlying network properties may change significantly. This means that while during certain periods of the call we will experience good call quality, there may be intervals with poor quality or even complete lack of voice reception. This raises the legitimate question of how to decide the quality of these calls. While using the ITU-T recommendations formulae we get a quality estimate, this merely averages call parameters ignoring important information. In order to work around these shortcomings, we propose the definition of *Extended MOS (E-MOS)*. This definition is based both on the original MOS and on the IPPM's `Type-P` packet described previously.

The main improvement of Extended MOS is the division of the voice stream in smaller segments and to perform call's quality computation on these chunks. Each one of these chunks can represent from a predetermined number of packets to a variable size talkspurts (Nomenclature obtained from [3] referring to continuous talk from one person).

There are many applications of this approach, for example E-MOS can be used for accurately billing the customer depending on the quality of the conversation. It also can be used by operators in order to know the current delivered call quality, and in the event of a network failure take actions, namely trigger a route change or load balancing, for guaranteeing the proper call quality. This is possible because E-MOS is able to report information in a short time scale.

Algorithm 1 `Type-P-MOS` pseudo code

Input: Packet[1..n], C {Packet's input stream, it might be unbounded. C is the used codec for transmission}
i = 1
t = $\mathbf{T_0}$
S = {} {Initialise S, it will hold the list of selected packets}
repeat
  $k \leftarrow F_p(Packet[i])$ {Selects the first packet of the stream provided it is a selectable `Type-P` Packet}
  if k **then**
    $S \leftarrow S \cup Packet[i]$
  end if
  i++
  $t \leftarrow getSendTimeStamp(Packet[i])$
until $t >= \mathbf{T_f} \ \vee \ i >= n$
Output: MOS(S, C)

A. Type-P-MOS

Let's define a singleton metric called `Type-P-MOS` with the following parameters:

- *Source*: Source IP address of a host.
- *Destination*: Destination IP address of a host.
- T_0: an initial time.
- T_f: a finish time.
- *C*: a voice codec from the list found in [13].
- *F*: a selection function defining unambiguously the packets taken from the stream selected for the metric. It takes two parameters, P, described below, and a packet. It will output the packet or `null` if the packet is not selectable.
- *p*: the specification of the packet type. This will define a F_P which holds the list of packets chosen for the metric.

Both T_0 and T_f form a time interval that determines the period of packet selection decided by `F`.

This metric is composed by One-Way Delay [10], IPDV [4], packet loss ratio [11] and the voice codec [13] used for the communication.

1) Algorithm: `Type-P-MOS` reports the MOS value obtained from the E-Model from the selected packets by F_p. The pseudo code for this metric follows in Algorithm 1.

The Input list of packets must have the sending and receiving timestamps of the packet, or in the case of packet loss the sending timestamp and a mark indicating packet loss.

B. Type-P-EMOS-*-Stream

`Type-P-EMOS-*-Stream` uses `Type-P-MOS` as a base for a new metric. Its parameters are:

- *Source*: source IP of a host.
- *Destination*: destination IP of a host.
- T_0: a initial time.
- T_f: a finish time.

Authorized licensed use limited to: UNIVERSITAT POLITÈCNICA DE CATALUNYA. Downloaded on May 13, 2009 at 11:49 from IEEE Xplore. Restrictions apply.

Algorithm 2 `Type-P-EMOS-*-Stream` pseudo code

Input: Packet[1..n], C {Packet's input stream, it might be unbounded. C is the codec}
t_{th} = {0,0} {Contains the packet's interval it can hold timestamps or packet counts}
i = 0
while t_{th} = F(Packets,t_{th}) **do**
5: {Fills up t_{th} with the time interval decided by `F`}
$mosArray[i] \leftarrow$
$\texttt{Type} - \texttt{P} - \texttt{MOS}(Packet[t_{th_0}, t_{th_1}], C)$
if ActionNeeded **then**
TriggerEvent($mosArray[i]$)
10: **end if**
$timeArray[i] \leftarrow t_{th}$
i++
end while
Output: mosArray, timeArray

- F: a selection function.
- p: the specification of the packet type to select. This will define a F_p which holds the list of packets chosen for the metric.

Depending on the selection function, the metric might have different behaviour. Hence, the `*` in `Type-P-EMOS-*-Stream`.

1) Selection function: `F` in this case will decide the intervals depending on p (F_p), between T_0 and T_f which are the proper thresholds for computing the `Type-P-MOS` value. Such function specifies the capture boundaries. A detailed description of selection functions is left as an important part of our future work. Initial possibilities:

1) `Type-P-EMOS-Periodic-Stream`: Regular non-overlapping time intervals, this computes MOS values periodically over time, regardless the contents of the voice transmission.
2) `Type-P-EMOS-Sliding-Stream`: Similar to `Periodic` but the time intervals overlap over time. This permits to keep a history of past events to avoid reporting independent MOS values.
3) `Type-P-EMOS-Talkspurt-Stream`: For this to work prior knowledge of the codec, silence detection algorithms and methods for payload examination of the traffic are needed.

2) Algorithm: This metric applies the `Type-P-MOS` metric to the packets contained in the limits expressed by T_0 and T_f. Pseudo-code for this operations is shown in Algorithm 2. The output is an array of n MOS values.

The algorithm is straight-forward, it selects the lower and higher boundaries of the packet stream, it computes MOS over that fragment. The system monitors whether the MOS is within valid boundaries, triggering the required action if needed.

For off-line processing when all the packets in the input stream have been processed, the `mosArray` and the `timeArray` are returned.

3) Metric results: As shown in the metric definition the output is an array of values, this array gives the voice quality over time. With this information it is possible to have accurate reporting of the status of the voice quality. This can be used by service providers to give feedback to the users about the delivered voice quality.

Some statistics definitions for `Type-P-EMOS-*-Stream`:

- `Type-P-EMOS-*-Mean`: Refers to the mean value of the $mosArray$ output. This value is the closer to the original MOS as will be shown later.
- `Type-P-EMOS-*-Std`: Is the standard deviation of the $mosArray$ output.
- `Type-P-EMOS-*-Percentile`: Given a percentile (`P`) value between 0% and 100% the value which has P% values below. This can be useful for outliers detection.
- `Type-P-EMOS-*-Median`: This metric is equivalent to the 50th percentile except when even number of values are returned, in that case the mean value between them is taken.
- `Type-P-EMOS-*-Minimum`: The minimum of all the `Type-P-MOS` values.
- `Type-P-EMOS-*-Maximum`: The maximum of all the `Type-P-MOS` values.

V. Validation

Once the metric has been presented this section is focused on the validation of the system, we also present the tests and the results to verify the behaviour of the proposed metric in a real scenario.

Due to space limitations, this section takes for validation purposes the `Type-P-EMOS-`*`Periodic`*`-Stream`, extending the validation to other metrics is straight-forward.

E-MOS supersedes original MOS. Since `Type-P-EMOS-*-Mean` can deliver the same call quality as MOS with a bounded error ($\pm\varepsilon$). Where ε is `Type-P-EMOS-*-Std` because MOS algorithm uses *Mean Delays* and *Packet Loss Probability* of the whole conversation. For more detailed information on MOS computation refer to [13].

A. Methodology

As discussed previously in the literature [23] testing is not a straightforward task. This section details the different techniques we have used for guaranteeing the soundness of the results presented later.

1) Capturing environment: The set of tests prepared for this paper use real applications (Linphone in this case), usually such end user tools are not suited for delivering detailed statistics about network information (i.e. per packet one way delays, packet losses), with this situation there is the need of complementary tools to perform such tasks.

A first approach could be the use of other tools to actively generate traffic which could resemble somewhat the actual voice traffic generated by the application. As seems obvious, in order to simulate the traffic flows is not a good approach

Authorized licensed use limited to: UNIVERSITAT POLITÈCNICA DE CATALUNYA. Downloaded on May 13, 2009 at 11:49 from IEEE Xplore. Restrictions apply.

given that the codecs used do not generate constant bit rate traffic, specially due to silence detection.

As a second option, there is the possibility to use available passive capturing tools (i.e. Ethereal, tcpdump...). The problem found with this approach is the need of computing one way delays and packet losses of the flows under test. This forces to set up two capture points, one in the source host and the second at reception. Moreover, this approach needs the development of an algorithm for flow detection and packet impairment at both ends from the trace files, which is inherently inefficient because the full packet payload must be captured and stored for later processing.

The problem with this second option is the (automatic) correlation between both traces. That's why we chose a third approach, which is using OreNETa [26]. This tool uses the `libpcap` library for capturing the packets, the difference with other tools using this library resides in the fact that it permits to specify different capture points simultaneously. OreNETa by using the mechanism described at [28] is able to automatically detect the flows and the packets, compute one way delays and packet loss ratios, such data is later processed to obtain statistical results.

2) Testbed: The testbed used for this work is composed by the two end-points and a Linux router. The goal is to keep it as simple as possible and to have control over the network behaviour using queueing mechanisms.

One of the main parameters needed for the metric are One Way Delays, to compute it it is mandatory to have the proper synchronisation on the equipment as the timestamps must be comparable. To do so we used NTP.

We enhanced the precision of the clocks by using several stratum-1 reliable time sources, specifically we used two separated GPS servers in our lab.

Another source of noise is the variability among the tests. Voice encoding and generated traffic might vary depending on the silence periods or the actual voice of the speaker. For avoiding incorrect results caused by this variability we used a prerecorded conversation together with Linphone which lasted for 4 minutes 16 seconds. The conversation was a standard dialogue between two persons talking English.

We decided to use the previously recorded couple of files (one for each direction of the dialogue) for all the tests, this way potential changes on the conversation would not affect our tests. To guarantee proper interpretation of the results we fixed the transmission Codec (C) to G.711. To transmit the voice while avoiding echoes we installed two sound cards on each computer, one for transmitting the prerecorded WAV file, and connected with an external cable to the line-in input of the other card, which was the one actually feeding the data to the VoIP application.

All this set up permits to automate the generation of test. We were able to capture each test separately and to repeat them as many times as we needed.

3) Network characteristics: As the testbed is set up on a local network there is no congestion is encountered. We introduced controlled sources of variation using `netem` to emulate different kind of network behaviour.

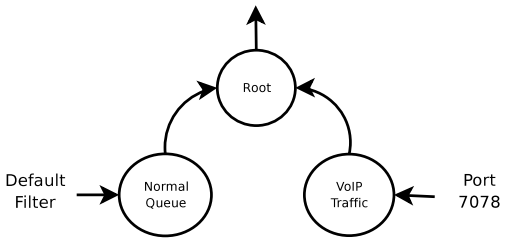


Fig. 1. Queue hierarchy

`netem` is a traffic control (`tc` [19]) mechanism available on current Linux Kernel (we used 2.6.15 in our tests), which permits to set up different network conditions in an easy way. A deep description of this software is out of the scope of this paper, here we will only describe the basic functionalities we used for emulating large one way delays, packet losses and high jitter.

Figure 1 shows the queueing hierarchy used for the tests. The key point in the proposed scheduling is the fact that the average Internet traffic doesn't get affected by the `netem` tweaked queueing. For doing so, we forced Linphone to use port 7078 and we filter all the UDP traffic outgoing from the machines in the specified port, this way the Linphone control channel is going through the standard Linux queueing mechanism.

Each performed test has different values for packet loss ratio, one way delay and jitter. Each change provides different conditions for computing E-MOS experimentally.

In one way delay we modeled (as `netem` permits) the delays around a normal distribution of the specified value, their details are discussed later.

B. Tests

We performed two different set of tests, first with constant network conditions and the second using variable parameters.

The first set of tests, along with its main characteristics are summarised in Table II. Both delay and losses are added in a controlled way. The delays are modified by a pseudo random jitter of 3ms in delay to have a more realistic environment.

The goal of this set is to prove that standard MOS gives accurate results as long as the network metrics are stable during the tests. That is, the packet losses and the one way delays are equally distributed along the whole conversation.

Each tests was repeated several times for achieving statistical soundness. Table II summarises the Delay and Packet Losses obtained from testing. The table shows the computed mean for each type of test.

We also performed tests with 0.1% loss ratio, but given the low packet rate of the voice flows the results are similar to the lossless case, thus are not shown in the table.

As it can be noted, some of the performed tests are not realistic on a real scenario, namely, 50% losses or delays bigger than 200ms (although as shown in [15] some applications

Authorized licensed use limited to: UNIVERSITAT POLITÈCNICA DE CATALUNYA. Downloaded on May 13, 2009 at 11:49 from IEEE Xplore. Restrictions apply.

TABLE II
MEAN DELAYS AND PACKET LOSSES

Set	Characteristics	Average	Std Dev
Test 1	Loss 50%	49.88%	0.8%
Test 2	Loss 25%	25.3%	0.3%
Test 3	Loss 10%	10.61%	0.1%
Test 4	Loss 5%	5%	$\sim 0\%$
Test 5	Loss 1%	1.4%	$\sim 0\%$
Test 6	Loss 0%	0%	0%
Test 7	Delay 500 (ms)	497.37	6.41
Test 8	Delay 300 (ms)	301.83	4.32
Test 9	Delay 100 (ms)	104.28	2.38
Test 10	Delay 50 (ms)	53.23	0.83
Test 11	Delay 0 (ms)	3.4	$1.9 \cdot 10^{-4}$

have bigger mouth to ear delays) but our goal is to highlight the improvement acquired by E-MOS over MOS.

In the second set of tests, the variation of delays was not constant, there was an increase in delays of 10ms each 10 seconds, starting with 1ms until 300ms of delay at the end of the test. Moreover, for having more variability, a jitter proportional at 10% of the delay value is forced. With this behaviour it is very easy to notice the inherent problems of the legacy MOS algorithm with only one value as result.

C. Results

The main focus of the paper is to show the improvement we obtain by using E-MOS over standard MOS. For this purpose here we present the results obtained from the two different testset.

1) Homogeneous network conditions: As described in the Tests section, the set of performed measurements treat separately packet losses and delays. This way it is possible to isolate each metric effect over the final call quality all over the test.

a) Packet Losses: Before studying packet loss effects on call quality two different aspects must be considered. First packet losses effects in the final MOS value is a work in progress as stated in ITU's reccommendation G.113 Appendix A [14]. Second the outcome of the results depend strongly on the codec. For this purpose we forced Linphone to use the G.711 Codec.

This tests with homogeneous network conditions have a twofold goal. On the one hand we validate the good results obtained standard MOS algorithm when network conditions don't change drastically over time. This highlights that MOS, as designed for circuit switched networks was a good approach, even if it should be adapted to the new network dynamics.

On the other hand we point out the improvement in reporting precision we obtain by using our E-MOS proposal.

Table III shows the mean values both of delay and loss for each testset with controlled packet losses. We used 1s, 3s and 5s boundaries for computing the parameters, the table shows the 1s case of `Type-P-EMOS-Periodic-Stream`.

TABLE III
MOS WITH CONTROLLED PACKET LOSSES (1S PERIODIC)

	Test 1	*Test 2*	*Test 3*	*Test 4*	*Test 5*	*Test 6*
Loss Ratio	49%	25%	10%	5%	1.4%	0%
Delay(ms)	0.9	4.1	4.2	4.4	5.1	3.5
MOS	1.3	1.78	2.68	3.24	3.87	4.05
E-MOS Mean	1.36	1.94	2.85	3.34	3.89	4.05
E-MOS Std	0.12	0.41	0.56	0.53	0.33	0.14

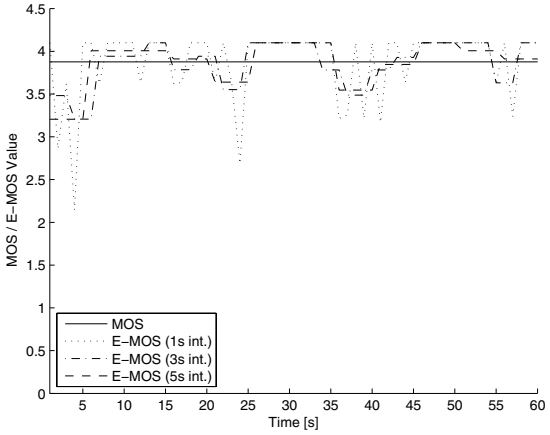


Fig. 2. E-MOS evolution (1% packet losses)

In the table, E-MOS Mean and E-MOS Std. refer respectively to `Type-P-EMOS-Periodic-Mean` and `Type-P-EMOS-Periodic-Std` metrics defined in section IV.

As it can be noted E-MOS Mean is similar to the MOS as the network conditions are kept during all the test. The results show that for having a reasonable minimum quality, losses should be kept below 1.4% that corresponds to *Some users satisfied* entry on Table I. With higher loss ratios MOS and E-MOS values are clearly below the threshold of admissible quality.

Figure 2 shows the evolution of E-MOS over the first minute for 1% losses test. The figure presents E-MOS computed for 1s, 3s, and 5s intervals respectively for `Type-P-EMOS-Periodic`. The MOS value of the whole conversation is also shown as baseline to which we can compare the other results.

With homogeneous network conditions, increasing the period of the metric tend to smooth the variability of the result. In lower timescales with the low packet rate of VoIP traffic the homogeneity is not preserved.

The figure illustrates the improvements brought by E-MOS. Where MOS reports a single static value, E-MOS delivers periodic feedback about the call quality. This can be used by operators to prove the service is being properly delivered, or it

Authorized licensed use limited to: UNIVERSITAT POLITÈCNICA DE CATALUNYA. Downloaded on May 13, 2009 at 11:49 from IEEE Xplore. Restrictions apply.

can even trigger the corresponding network control entities that can provision extra resources, or change the billing algorithm as decided on the customer's contract.

An interesting outcome of the analysis of E-MOS over the conversation is the amount of time an user feels good quality while having the conversation. Figure 3(a) illustrates that as the packet losses increase it's quality decreases as expected, up to the point of having almost all the conversation under minimum quality conditions for the 50% and 25% loss.

The case where everyone is satisfied doesn't have any occurrence as the G.711 codec has a maximum theoretical value if 4.11 which is below the 4.34 limit for this entry on Table I. The histogram also highlights that when there are 10% or more losses then more that in 50% of the call time is very difficult or impossible to understand the conversation (e.g. MOS below 3.10).

b) Delay: Delay analysis is performed in a similar way as the loss. Table IV shows the obtained mean delay, MOS value and E-MOS mean. There is no Loss entry because given the good conditions of the network no losses occurred during the tests at network level. Therefore, the column related to 0ms delay is also omitted because the results are the same as in loss 0%.

TABLE IV
MOS WITH CONTROLLED ONE WAY DELAYS (UNITS IN MS)

	Test 7	*Test 8*	*Test 9*	*Test 10*
Delay	497.37	307.37	104.28	53.23
MOS	2.29	2.96	3.97	3.98
E. Mean	2.23	2.99	3.84	4.04
E. Std	0.32	0.36	0.33	0.01

Related to packet losses, the testbed computes *Network* losses due to Network problems. During the tests there were some losses at application level, this is because of the real-time nature of the conversation.

In Figure 3(b) it can be seen the more deterministic effect of one way delays over the call quality. This happens because the network homogeneity for one way delay is preserved regardless the considered timescale. While packet losses are discrete and have bigger impact at small time scales.

Another important implication is that one way delays bigger than 100ms stimulate a considerable conversation degradation, as it drops from *Satisfied* to *Some Satisfied* (see Table I).

2) Variable network behavior: The second testset instead of keeping homogeneous network conditions is focused on studying the effects of increasing delay over MOS and E-MOS during the conversation. This time the difference between both metrics is much bigger as MOS does not react properly to high network variability.

Figure 4 shows the E-MOS value computed on 1s and 5s intervals, and the overall MOS value. There is a threshold at 250 ms delay where E-MOS reaches the lower bound and renders the conversation not understandable.

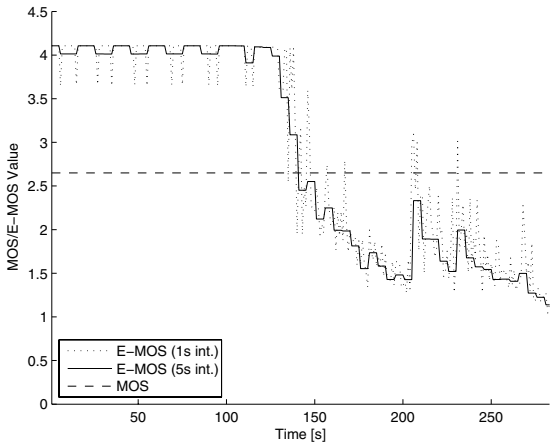


Fig. 4. E-MOS Progressive Test

On the other hand, MOS equals to 2.6, which means that almost all users are not satisfied about the quality, while in reality it was perfectly good during 50% of the test duration. This difference is more noticeable with E-MOS results. It reports a mean of 1.97 with a standard deviation of 1.2 meaning that such values are not statistically significant.

This high variability is not common in the Internet, but highlights the point that with high jitter, or important changes in network conditions, MOS is not a proper metric for voice quality measurement. While enhancing it with E-MOS permits to differentiate clearly which parts of the conversation are good or not.

VI. CONCLUSIONS AND FUTURE WORK

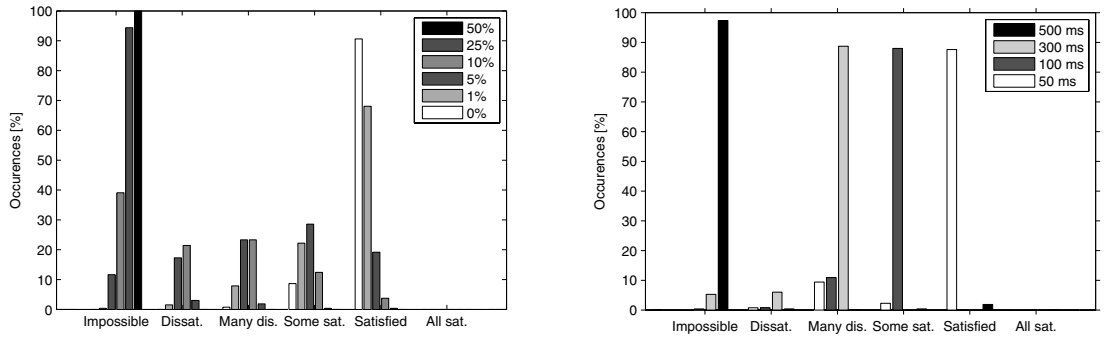
This paper had two main purposes, first to study existing quality reporting tools and develop a new metric derived of currently existing methods. Second to study, under several network conditions the effects and differences between the developed metric and the original MOS approach. Such differences highlight the need and usefulness of our proposal for proper quality reporting in short timescales.

This metric has been developed using definitions and methodology of ITU-T and IPPM. From ITU-T the MOS definition and implementation has been adapted to suit the new VoIP paradigm in the Internet. Regarding IPPM the low level network metrics used for measurement network performance (one way delay and losses) have been used together with IPPM's methodology for defining new metrics with the definition of `Type-P-MOS` and `Type-PEMOS-*-Stream` metrics.

In order to validate E-MOS, two different set of tests have been performed. The first round shows that under constant network conditions over time it is valid to use the classical MOS approach.

The problems with this single value metrics arises when the network conditions change over time, which in currently

Authorized licensed use limited to: UNIVERSITAT POLITÈCNICA DE CATALUNYA. Downloaded on May 13, 2009 at 11:49 from IEEE Xplore. Restrictions apply.



(a) Packet losses

(b) One Way Delay

Fig. 3. User satisfaction

available networks is more than likely to happen. The solution for proper voice quality reporting is to use the proposed E-MOS metric, this enables potential operators or service providers to keep a more detailed track of the delivered quality over time.

There is, however, some work which has been left for further study. The defined metric can be improved by defining more selection functions (`S`) in detail, specifically talkspurt detection.

More important, when the goal is to have a Quality of Service enabled environment, is to have guaranties that the quality is being delivered, with E-MOS it is possible to know such quality accurately and act upon the network. Further study is needed for enhancing the metric with prediction capabilities (i.e. using Kalman Filters can prove useful).

Finally from the testing point of view, it has been limited to modifications in one way delay and packet losses, but another important metric to take into account is the delay variation, which usually is the cause of buffer underruns that cause cuts in the conversation.

REFERENCES

[1] IP Performance Metrics (IPPM) *http://www.ietf.org/html.charters/ippm-charter.htm*, January 1990.

[2] Alexander Raake. *Speech Quality of VoIP*. Wiley, September 2006.

[3] Catherine Boutremans and et. al. Impact of link failures on voip performance. In *International Workshop on Network and Operating System Support for Digital Audio and Video*, 2002.

[4] C. Demichelis and P. Chimento. IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). RFC 3393, November 2002.

[5] A. Cabellos-Aparicio, R. Serral-Gracià, L. Jakab, and J. Domingo-Pascual. Measurement Based Analysis of the Handover in a WLAN MIPv6 Scenario. In *Passive and Active Measurements 2005, LNCS 3431*, pages 207–218, 2005.

[6] Hui Min Chong and H. Scott Matthews. Comparative analysis of traditional telephone and voice-over-internet protocol (voip) systems. In *IEEE International Symposium on Electronics and the Environment*, 2004.

[7] R. G. Cole and J. H. Rosenbluth. Voice over ip performance monitoring. In *ACM SIGCOMM Computer Communication Review, Volume 31, Issue 2*, 2001.

[8] Falko Dressler. Advantages of voip in the german research network. In *IEEE - High Speed Networks and Multimedia Communications*, 2002.

[9] [IST] EuQoS - End-to-end Quality of Service support over heterogeneous networks - http://www.euqos.org/, September 2004.

[10] G. Almes and S. Kalidindi and M. Zekauskas. A One-way Delay Metric for IPPM. RFC 2679, September 1999.

[11] G. Almes and S. Kalidindi and M. Zekauskas. A One-way Packet Loss Metric for IPPM. RFC 2680, September 1999.

[12] ITU Inc. International Telecommunication Union: *http://www.itu.int*.

[13] ITU-T Recommendation G.107. The E-model, a computational model for use in transmission planning, 03/2005.

[14] ITU-T Recommendation G.113. Transmission impairments due to speech processing, 02/2001.

[15] Wenyu Jiang, Kazuumi Koguchi, and Henning Schulzrinne. QoS Evaluation of VoIP End-points. In *Passive And Active Measurements Workshop*, 2003.

[16] Wenyu Jiang and Henning Schulzrinne. Analysis of On-Off Patterns in VoIP and Their Effectg on Voice Traffic Aggregation. In *IEEE International Conference on Computer Communication Networks*, 2000.

[17] Wenyu Jiang and Henning Schulzrinne. Speech Recognition Performance as an Effective Perceived Quality Predictor. In *Tenth IEEE International Workshop on Quality of Service*, 2002.

[18] A. Lakaniemi, J. Rosti, and V. I. Räisänen. Subjective VoIP speech quality evaluation based on network measurements. In *IEEE International Conference on Communications ICC*, 2001.

[19] Linux advanced routing and traffic control - http://lartc.org/.

[20] Gao Lisha and Luo Junzhou. Performance Analysis of a P2P-Based VoIP Software. In *International Conference on Internet and Web Applications and Services/Advanced International Conference*, 2006.

[21] J.M. Lozano-Gendreau, Antoun Halabi, Maya Choueiri, and Valery Besong. VoWF (Vo-IP over Wi-Fi). In *IEEE Electronics, Communications and Computers*, 2006.

[22] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for IP Performance Metrics. RFC 2330, May 1998.

[23] Vern Paxson. Strategies for sound internet measurement. In *Internet Measurements Conference*, 2004.

[24] R. Serral-Gracià, L. Jakab, and J. Domingo-Pascual. Out of order packets analysis on a real network environment. In *2nd Conference on Next Generation Internet Design and Engineering*, 2006.

[25] A. Clark T. Friedman, R. Caceres. Rtp control protocol extended reports (rtcp xr). RFC 3611, 2003.

[26] Mihai Vlad, Ionuţ Sandu, René Serral-Gracià, and Abel Navarro. OreNETa 2.0 - http://www.ccaba.upc.edu/oreneta, 2004.

[27] Wei Wang, Soung Chang Liew, and Victor O. K. Li. Solutions to performance problems in voip over a 802.11 wireless lan. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 54(1), 2005.

[28] Tanja Zseby, Sebastian Zander, and Georg Carle. Evaluation of building blocks for passive one-way-delay measurements. In *Passive and Active Measurements Conference*, 2001.

Authorized licensed use limited to: UNIVERSITAT POLITÈCNICA DE CATALUNYA. Downloaded on May 13, 2009 at 11:49 from IEEE Xplore. Restrictions apply.