Improving QoE of SIP-based Automated Voice Interaction in Mobile Networks

Jakub Dolezal, Lukas Kencl
Department of Telecommunication Engineering, Faculty of Electrical Engineering
Czech Technical University in Prague, Czech Republic
{jakub.dolezal, lukas.kencl}@rdc.cz

***Abstract*—Mobile voice-assisted services are currently experiencing strong growth. However, occasionally low real-time quality of service within mobile networks could have significant negative impact on quality of experience of users interacting with automated voice services. Latency may grow to unacceptable levels and speech recognition and synthesis might suffer. We present a methodology of mitigating such effects by monitoring the immediate connection status and adapting various parameters of the SIP communication setup (buffer size, codec) in response, thus radically improving user experience. We demonstrate practical usability by implementation and testing in a real mobile network and by performing multiple test scenarios when interacting with a state-of-the-art automated voice platform.**

I. Introduction

Recent years have brought huge advances in the areas of mobile internet access, Voice-over-IP (VoIP) and in voice-driven services that have exploited the increasing quality of automated speech recognition (ASR) and synthesis (text-to-speech, TTS). The combination of all these technologies has a strong potential to deliver a vast array of benefits and challenges – speech is suitable for e.g. querying commands when using mobile terminals in outdoor environments, eliminating the necessity to interact with a limited graphical user interface (GUI). Mobility allows users to request information anywhere and any time. However, mobility also has a negative impact on Quality of Service, crippling playback of both synthesized speech and user's speech to be recognized, which decreases the Quality of User Experience. Such a situation poses a challenge for deployment of voice-driven services, As remedy, a VoIP call carried over the mobile internet may improve the accuracy of speech recognition by e.g. using a high-quality codec.

Quality of Service (QoS) is a set of metrics describing qualities of network connection between the client and server. Conversely, Quality of User Experience (QoE) indicates how a service or an application is usable from the Human-Computer interaction point of view. The relation between QoS and QoE in mobile networks is not well understood. The majority of related work either describes simulated and measured characteristics of mobile networks or introduces solutions for adapting VoIP sessions to lower QoS.

The main contributions of this work is to address the QoS-QoE problem systematically in three steps: first, real-life traffic in a mobile network was measured and evaluated, next a novel feedback-driven algorithm for incremental improving is proposed and finally, results improvement of QoE are discussed. To this end, a SipDroid [1] SIP (Session Initiation Protocol) client for Android-enabled mobile phones was enriched by a comprehensive measurement functionality, as described in Section IV, coupled with the feedback algorithm. Taking SIP calls as an example of VoIP or – more generally – an example of a service demanding real-time response, we propose the count of dialog interruptions, caused by lower QoS, as major QoE metric. SIP has critical requirements on latency and goodput. With SIP being a commercial threat to conventional voice services by mobile carriers, low-priority policy is often assigned to its QoS in mobile wireless networks. We have carried out measurements to demonstrate the negative impact of lower QoS onto QoE. The passive QoS measurements monitor huge amount of user-data with useful payload, without the need to probe the network. Moreover, additional metrics are collected to discover relations among QoE, signal coverage and location.

II. Related Work

The feedback-driven algorithm proposed in this article has risen from multiple fields of study.

Larger packet size has a negative impact on both one-way delay and loss rate in 3G networks [2]. This calls for switching to a lower-bitrate codec when facing a low-quality network. QoS measurements of Skype voice calls using in real UMTS network were described in [3], including the impact of various codecs. According to [4], user mobility implies lower quality of service, e.g. 10% of HSUPA and 6% of HSDPA packets in the drive test exceed 150 ms delay [5] (a limit of good quality). The same research team delivered the QoSMeT tool [6] for measuring one-way QoS performance in 3G networks. The tool currently runs on MS Windows only, omitting widely used mobile phone operating systems. The measured QoS metrics are limited to the network connection only, not taking into account signal strength, user location or detection of possible interruptions in the incoming audio signal. Automatic estimation of Pseudo-Subjective Quality Assessment (PSQA) is proposed in [7], exploiting codec bitrate, one-way delay, jitter, loss rate and other network metrics.

Although various VoIP/SIP quality models exist, they are limited by different aspects. Mean Opinion Score (MOS) [8] was designed to be evaluated by humans only. ITU-T E-model exploits packet delay, loss, room noise etc. [9], however it is too complex and omits interruption count and missing metric.

978-3-901882-48-7

An adaptive strategy of switching the codec and the transport protocol was examined as an improvement mechanism in [10]. A special VoIP client was developed for measurement purposes. Assessment of effects of packet loss on speech quality in VoIP is investigated in [11] and expressed using various algorithm for expressing speech quality. An extended E-Model for prediction of speech quality in VoIP based on jitter and packet loss is investigated in [12].

A voice/video over IP framework for Android-enabled mobile devices was developed in [13], including measurement of the most important QoS metrics. However, no improvement for corrupted QoS was designed. Also, this is a largely server-based solution, while our method is solely client-based, as it stands in better position to determine immediate conditions. Similar way is presented in [14]. A tool for monitoring QoS and QoE is described in [15], deployed in network backbone.

Quality E-model for monitoring QoE in VoIP services is presented in [16] along with experimental results. An empirical study of impact of packet loss on VoIP calls in [17] evaluates HSDPA and HSUPA wireless technologies and suggests improvements to operator networks. Impact of packet-loss burstiness on speech quality is examined in [18].

Improvements of jitter buffer on QoS are discussed in [19]. Jitter buffers modeled by Markov Modulated Poison Process are analyzed in [20]. QoE of VoIP accessible through mobile networks are considered from coding perspective in [21]. Speech quality models based on delay and jitter are assess in [22] inspecting various types of network.

III. Impact of Lower QoS on QoE in 3G Networks

Measurements were carried out within a 3G mobile network to reveal factors that may cause QoS degradation.

A. Preliminary Measurements

An application was set up on a mobile phone, downloading 2MB and uploading 1MB of data and executing multiple *ping* commands to retrieve the round-trip time (RTT). Measurements of throughput and delay were taken over two 30-minute walks in a crowded district: at 11am and at 3pm. Averaged measured data over both walks are in Fig. 1.

Two SIP codecs are widely supported: the hi-quality Pulse Code Modulation codec (PCMU), defined by the ITU-T G.711 [23], and the low-quality GSM codec, defined by 3GPP [24]. PCMU claims 128 Kbps goodput (considering 2 channels of full-duplex connection). Accounting for all protocol headers, 160 Kbps throughput is required. This condition was met over the measurements, as the minimal download rate was 746 Kbps and the minimal upload rate 357 Kbps.

However, RTT values indicate that delay is a drawback. RTT does not correlate with throughput. Although the forth and back components of RTT are asymmetric, for simplicity let us assume that One Way Delay (OWD) is half of RTT, thus mean RTT equals 127 ms and mean OWD equals 63.5 ms, quite a high packet delay encountered in a SIP call. The outcome of preliminary measurements, even if not representative, indicates potentially significant degradation of the delay part of QoS.

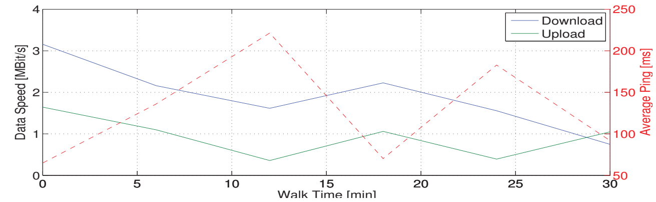


Fig. 1: Averaged values of 3G QoS measurement performed in morning and afternoon. While data speed is fully sufficient for SIP calls, average ping values indicate severely high latency with possible negative impact to QoE.

B. Test Measurements Setup

A server-based text-to-speech application has been made accessible through a SIP call. A total of 72 calls were carried out using an extended SipDroid client (described in Sec. IV). Each call lasted 200 seconds employing the PCMU codec so providing 10^4 packets, total number of $72 \cdot 10^4$ packets were examined. Two scenarios have been prepared to investigate QoS and QoE under different circumstances:

Walk, lasting 30 minutes and covering a university campus along with a crowded square, streets and a metro station (the same as in preliminary measurements);

Tram Drive, lasting approximately 15 minutes and including both the city center and outskirts.

Scenarios were executed multiple times in morning, noon and evening. QoS is represented by the delay and jitter metrics of the Real-time Transport Protocol (RTP) [25] packets carrying the PCMU payload. Delay values are computed:

$$\begin{aligned} delay_{i,a} &= time_i - (sequence_i \cdot packet\_interval) \\ delay_{i,r} &= delay_{i,a} - delay_{i-1,a}, \end{aligned} \tag{1}$$

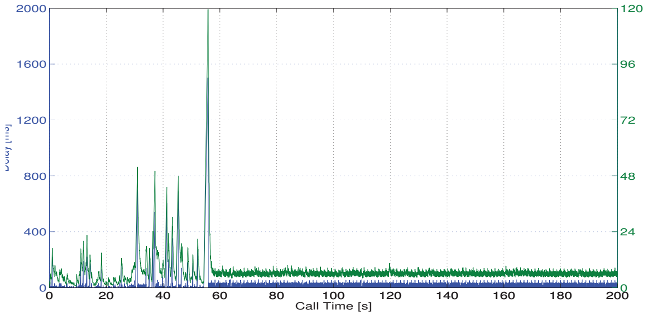
where $time_i$ is the relative time of arrival of the i-th received packet counting from the call start, and $sequence_i$ is the packet sequence number from the RTP header. The $packet\_interval$ constant defines a codec-dependent period of sending the next packet, e.g. for PCMU it is equal to 20 ms. The first equation expresses the absolute delay, $delay_{i,a}$, of the i-th packet. This value includes accumulated delays of previous packets. To overcome this, relative delay $delay_{i,r}$ is used further as the difference between absolute delay of i-th packet and previous $(i-1)$-th packet.

Jitter is the statistical variance of packet interarrival times [25], estimated using a simplified formula:

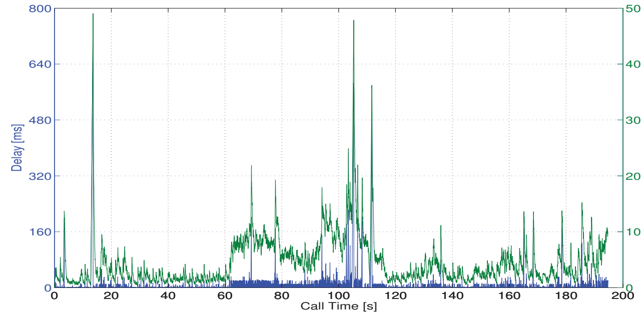
$$\begin{aligned} diff_i &= |(time_i - time_{i-1}) - (ts_i - ts_{i-1})| \\ jitter_i &= jitter_{i-1} + \frac{(diff_i - jitter_{i-1})}{noise\_reduction}, \end{aligned} \tag{2}$$

where ts_i is the timestamp of the i-th packet. First, a difference of relative transit times between two fellow packets is computed. Next, change of arrival times is computed and divided by the $noise\_reduction = 16$ constant (as recommended in [25]), to reduce the influence of random fluctuations.

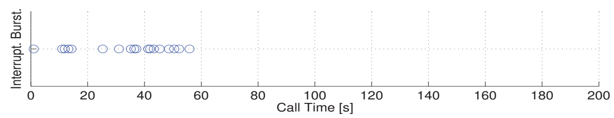
QoE interruption count was measured counting interruptions longer than 150 ms only, a boundary value for high conversational quality of voice calls according to ITU [26].



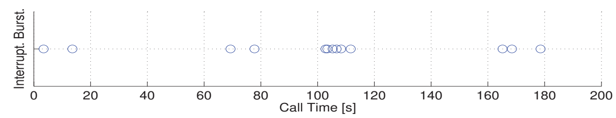
(a) Walk QoS



(b) Tram Drive QoS



(c) Walk QoE



(d) Drive QoE

Fig. 2: Representative results for the Walk and Tram Drive scenarios throughout a SIP call lasting 200s. QoS consisting of packet delay and jitter metrics is outlined for Walk (a) and Tram drive (b). QoE represented by interruption burstiness of both scenarios differs greatly: one long burst is typical for Walk (c), conversely, multiple short bursts are typical for Tram drive (d).

C. Results

Measurement results for both the *Walk* and *Tram Drive* scenarios are outlined in Fig. 2. The key difference between the scenarios is the length and density of bursts. For *Walk*, bursts exhibit length of about 12 seconds and high density. Conversely, for *Tram Drive*, the sole burst is 8 second short and sparse accompanied by multiple isolated interruptions. Signal strength or handovers exhibit no impact on QoE. Analysis of the *Walk* scenario supports a hypothesis of QoS being decreased by a number of active mobile phones nearby.

Interruption burstiness is analyzed for all calls in Fig. 3. Detected bursts are matched by predefined clusters lasting from 4 to 24 seconds with density from 0.5 to 1.5 interruptions per second. Mean share reflects the average count of a cluster within a degraded call and suggest that interruption burstiness

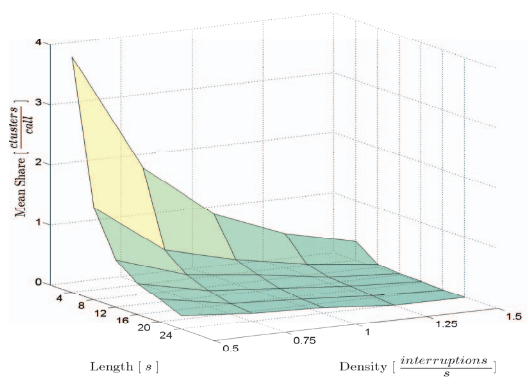


Fig. 3: Interruption clusters per length and density. Mean share indicates probability of cluster presence in a SIP call facing degraded QoS, calls with no interruptions are not considered. The most common clusters have length up to 12s and density up to 1 interruption per second, however, less frequent longer clusters with higher density include majority of interruptions.

– either 12 seconds long with density 0.5 interruptions per second or length 8 seconds long with density 0.75 or 4 seconds short and density 1.0. – causing lower QoE can influence almost every call lasting 200 or more seconds. We argue that a burst has a higher negative impact on QoE than a single interruption within several seconds of clear audio.

Loss of signal happened over the course of the measurements too. Hence, a new *missing* metric is introduced:

$$missing_j \quad = \quad \min\left(\frac{rate - count_j}{rate}, 1.0\right) \tag{3}$$

The *missing* metric is collected per time-frame j based on the count of received packets that is evaluated with known packet rate. The $rate$ is constant for a codec and can be computed as $\frac{1000ms}{packet\_interval}$, e.g. for the PCMU codec $packet\_interval$ is 20ms. Over 1000ms, amount of 50 packets is received for analysis. The metric can yield a value grater than 1.0 if in time-frame $j-1$ packets have been delayed and arrive in time-frame j, which is limited at 1.0 to prevent influence from previous time-frames. Percentiles for metrics are shown in Tab. I. The peaks have major impact on QoS, conversely the mean values for all metrics are close to 0.

No late packets were observed arriving out of sequence order. A total number of 7 hops were detected along the packet path using the Traceroute tool. Loss of packets was mostly 0. Rare losses were correlated with the *missing* metric.

TABLE I: Percentiles of measured delay, jitter and missing. For all the metrics, the 80th percentile is lower than 1ms or 1% respectively. A significant rise starts at the 99th percentile.

	Delay [ms]	Jitter [ms]	Missing [%]
Pct. 99.5	41.12	25.55	83.44
Pct. 99.0	31.28	18.78	66.39
Pct. 95.0	15.18	11.20	9.19

IV. Adaptive Speech-Droid (A-Droid)

An extension to the popular SipDroid [1] client, A-Droid is designed as a modified control system consisting of the measurement, controller (merged compute and compare), adaptation (correct) and playback components (see Fig. 4). It is a *closed-loop*, *feed-forward* (lower QoS measurements may serve as early warning to QoE drop), *discrete* (executed once per time window), *linear*, *multi-input-single-output* (the error score computed by the controller is a linear combination of multiple variables) and *non-proportional* (scale of adaptation is not influenced by the error score size, instead constant-driven Additive Increase – Multiplicative Decrease (AIMD) [27] adaptation ensures permanent convergence towards the optimum) *control system*. Note that AIMD is also successfully used in TCP congestion avoidance.

A. Measurement Component

The following data are collected throughout the SIP call:
QoS – the state of the 3G network: network type, sequence number, packet interval, delay, jitter and lost and late counts;
QoE – the audio interruption count provided by the Playback Buffer (see Sec. IV-D);
Signal coverage consisting of cell IDs and signal strength. Collected cell IDs are then matched to context data (e.g. location or operator) using an online community database;
Location including latitude, longitude, accuracy and provider: either GPS or Network.

B. Control Component

The controller's logic is executed in a loop which each time window computes a `score` based on the QoS and QoE metrics. Its assessment determines the adaptation executed. The `tabu` timer, preventing excessive adaptation, is decreased every cycle.

The score is computed by Algorithm 1 once per time-window of 1s, so 50 samples are collected to reach a decision of possible adaptation. It is up for discussion whether 50 samples are adequate, however experimenting with other values was out of the scope of this paper.

Algorithm 1 Error `score` computation based on averaged QoS and QoE metrics collected over a time window. A low-pass filter is applied to include past scores. Score range is $[0, 1]$ where higher value means lower QoS and QoE.

for each time window j of a call **do**

$$D_j \leftarrow \frac{\sum delay_{j,i}}{count_j \cdot delay_{Max}}$$

$$J_j \leftarrow \frac{\sum jitter_{j,i}}{count_j \cdot jitter_{Max}}$$

$$M_j \leftarrow missing_j$$

$$N_j \leftarrow \frac{\sum interrupt_{j,i}}{interrupt_{Max}}$$

$$score_j \leftarrow \frac{D_j + J_j + M_j + N_j}{4}$$

$$score_j \leftarrow \alpha \cdot score_j + (1 - \alpha) \cdot score_{j-1}$$

end for

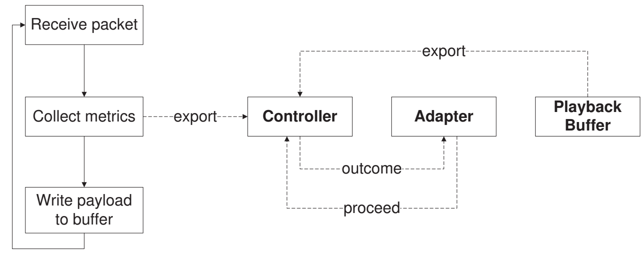


Fig. 4: The A-Droid control system collects the 3G-network QoS and the Playback-Buffer QoE metrics, computes an error score and determines the adaptation to improve QoE.

Four metrics are used as input to score computation:
1.-2. $delay_{j,i}$ and $jitter_{j,i}$ of the i-th packet received in the j-th time-window. Both metrics are averaged to provide overall information on QoS within the last window, in which $count_j$ packets were received. Rise of delay might cause just a single interruption or it might signal a whole burst of interruptions. Delay peaks and bursts are often preceded by high jitter, which serves as another signal. If adapting properly, delay impact on interruptions could be reduced;
3. $missing_j$ is the share of missing packets in the j-th window, as defined in Eq. 3. This metric rises when overall QoS is poor or signal has been lost, which cannot be detected another metric until an interruption occurs, since delay or jitter are 0;
4. $interrupt_{j,i}$ is i-th interruption in the j-th window. This metric indicates poor level of QoE, however its rise occurs when it is too late.

First, an average value of each metric is computed from fresh data measured in j-th window. Next, averaged values are normalized to [0,1] by division by a maximal constant related to each metric, defined as double the 99.5th percentile of averaged values from all measurements (see Table II), to be sufficiently high but to prevent misrepresenting influence of outliers. Missing does not need to be normalized since the metric itself is defined on scale [0,1].

Next, all metrics are combined using equal weights. Finally, a low-pass filter with parameter $\alpha \in [0, 1]$ is applied to take into account scores computed in previous windows. After `score` is computed, its value is assessed against a $threshold = 0.1$, i.e. `score` $\geq threshold$ is evaluated as adaptation indicator for the control system.

C. Adaptation Component

Based on `score` assessment, the Controller chooses and executes the appropriate adaptation scenario (see Fig. 5):
Buffer Resize is an Additive Increase – Multiplicative Decrease (AIMD) function to dynamically adjust the buffer size.

TABLE II: Max constants used in Algorithm 1, defined as doubles of the 99.5th percentiles of per-second means.

Metric	Delay [ms]	Jitter [ms]	Interrupt [%]
Pct. 99.5	33.197	25.805	1.222
Max constant	66.395	51.610	2.443

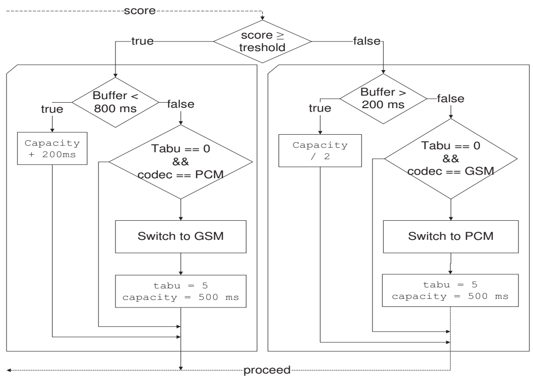


Fig. 5: Adaptation is launched according to `score` assessment. If `score` $\geq threshold$, the buffer is enlarged, or codec is swapped if buffer limit is reached. If `score` $< threshold$, buffer size is halved or codec is swapped to PCMU.

If `score` $\geq threshold$ (indicating worse QoS+QoE), it is increased by 200 ms, so that enough packets are received to assemble an uninterrupted chunk of playable audio. Conversely, when `score` $< threshold$, buffer size is decreased by half. If buffer size reaches its limit (minimum: 100ms, maximum: 1000ms), a *Codec Swap* is executed instead, if possible.
Codec Swap switches between the PCMU and GSM codecs. If the maximum (minimum) buffer limit is reached, the call is switched to the GSM (PCMU) codec, respectively. Switching to GSM codec decreases bitrate so it is necessary to lower packet length to improve QoS. Conversely, switching to PCMU increases bitrate, so it is possible to improve quality of transferred audio. After the swap, buffer size is set to 500 ms – a half of the maximum limit. Swapping the codec is a costly operation requiring call re-initialization (approximately 2s) and causing an interruption. Fortunately, this is handled by the SIP server, without impacting the text-to-speech application. To prevent frequent swaps, the `tabu` period is used to prevent a swap earlier than 5 seconds after the previous one.

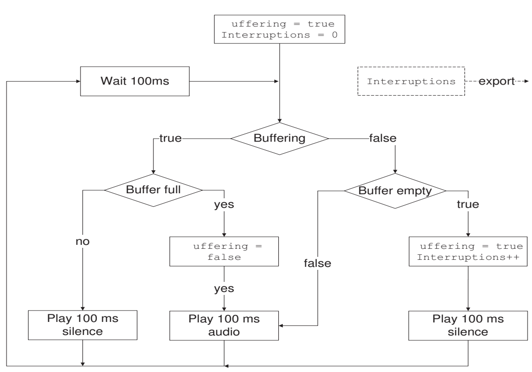


Fig. 6: The *Playback Buffer* stores payload of received packets. Periodically, depending on buffer status, it either plays 100ms of audio or of silence. In the latter case, the `Interruptions` counter is incremented and exported to the *Controller*.

D. Playback Buffer Component

The Playback Buffer is built as FIFO queue with synchronized access. Audio is produced each time a packet arrives and its payload is appended at the end of the queue. Audio consumption is driven by a loop (see Fig. 6), directed by a counter of accumulated audio and a boolean buffering state:
1. When not buffering or accumulated enough amount of audio according to buffer size, a constant amount of 100 ms of audio is consumed and played. If buffering or no enough data is available, 100 ms of silence is played;
2. After playing of audio or silence, go to 1.

When no audio is available and buffering has started and silence is played, an internal counter of interruptions is incremented and its value is exported to the Controller.

V. Performance Evaluation

Performance of A-Droid was simulated using all the measured SIP calls to verify correctness of score computation and execution of adaptation. The measurement module enables to run the simulations exactly like being run in a mobile phone. This allows evaluation of multiple settings of the α filter parameter and comparison of interruption counts with the measurements without adaptation. We have not used the codec swap, this option was not verifiable in a simulation.

Averaged improvements for both scenarios are outlined in Fig. 7: *Walk* without the adaptation experiences 16 interruptions, with adaptation enabled, count of interruptions is lowered to 9, resulting into improvement of 44% ($\alpha = 1.00$). *Tram Drive* then shows improvement from 13 to 8 interruptions (38%). Overall improvements are depicted in Fig. 8. Improvements for both scenarios showing boundaries and average values are in Tab. III. Necessity of adaptation for Tram Drive is higher than for Walk. Impact to both scenarios is equal with minor differences only.

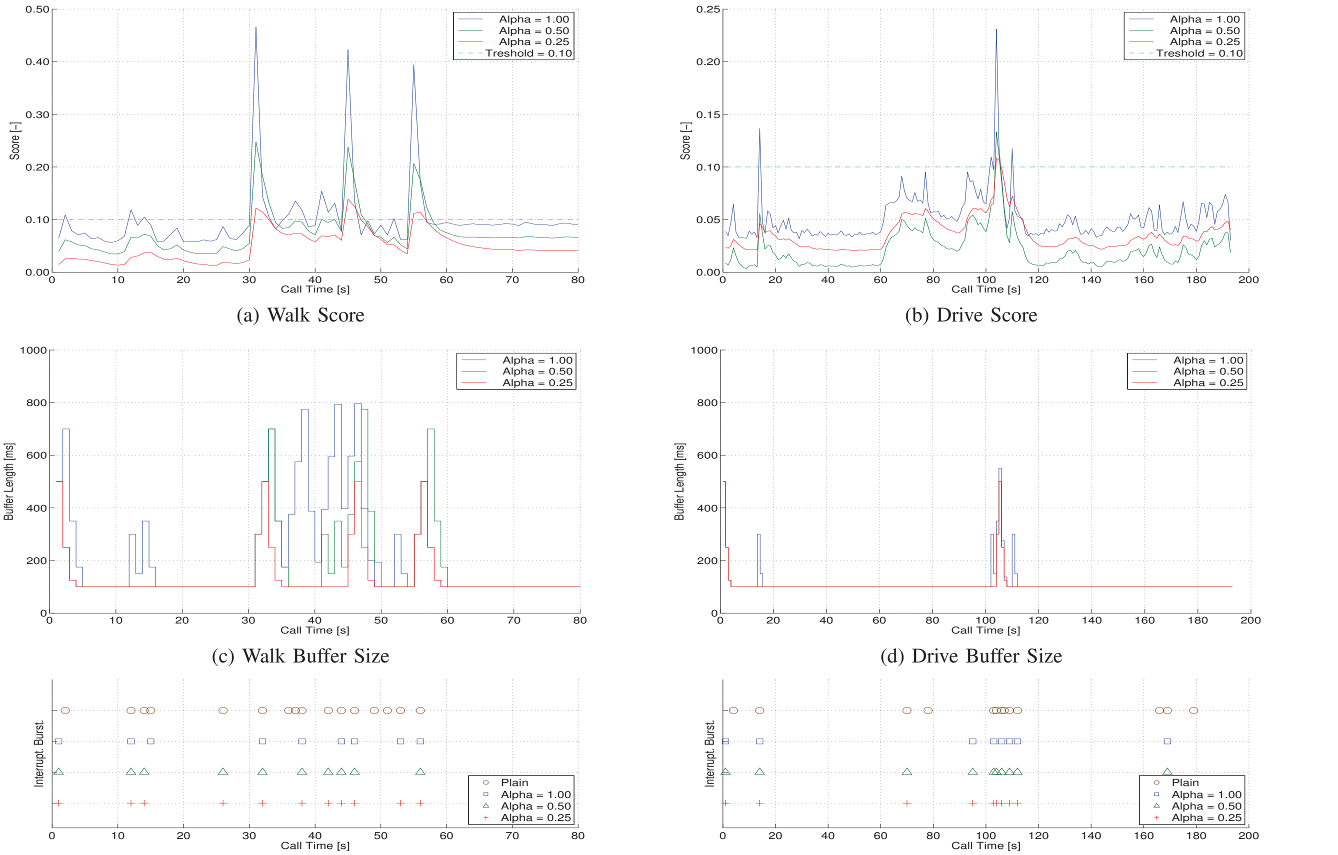
The evaluation discovered the importance of the filter parameter α for quick response to dynamic changes of QoS when $\alpha = 1.00$. However when facing slow change of QoS, $\alpha = 0.50$ is more adequate, accumulating previous score values. Generally the main benefit of A-Droid is the capability to adapt when approaching long bursts of interruptions. When a single interruption or sparse interruption burstiness is encountered, the A-Droid is limited in its reaction to the quick degradation of 3G, as it is difficult to predict.

VI. Conclusion and Future Work

We have presented a measurement-based study of QoS and QoE in contemporary mobile wireless networks, focusing on an automated voice-interactive service. We show that latency

TABLE III: QoE improvement measured over the fraction of calls facing QoS degradation. Mean, Min and Max represent the fraction of interruptions eliminated, per these calls.

	Calls degraded [%]	Mean [%]	Min [%]	Max [%]
Walk	67.50	54.57	22.22	83.87
Tram Drive	78.13	52.78	30.77	80.77



(a) Walk Score (b) Drive Score

(c) Walk Buffer Size (d) Drive Buffer Size

(e) Walk QoE Improvement (f) Drive QoE Improvement

Fig. 7: Scores for various window sizes for the Walk (a) and Tram drive (b) scenarios. Adaptation outcomes, represented by buffer sizes, are depicted for Walk (c) and Tram Drive (d). Finally, QoE improvement is represented by interruption burstiness for Walk (e) and Tram Drive (f). Charts showing buffer size (c) and interruptions (e) of Walk have a shorter time axis for better readability of improvement, since the remaining call time contains no interruptions.

has major impact on the most significant aspect of QoE - the number of interrupts in the speech-based interaction. To this end we have designed and implemented a simple adaptive algorithm on the mobile client, which adjusts the playback-buffer size according to the immediate measured QoS and is

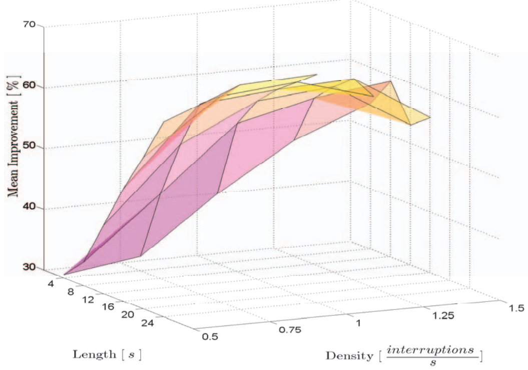


Fig. 8: Improvement averaged per cluster. The highest improvement is reached for longer clusters with higher interruption density.

especially effective in preventing bursts of interruptions. We show that using such technique, on average close to 53% and up to 84% of interrupts can be avoided.

With the ascendancy of mobile clients and cloud-based voice-interactive services, yet the ever-unpredictable QoS in mobile access networks, such adaptive solutions will be necessary for maintaining reasonable level of QoE and customer satisfaction. While adaptation has been shown a necessity, the optimal adaptation algorithm is certainly subject to future work and may eventually include not only the network performance information, but also contextual data such as location, time, number of concurrent users or speed of movement. Furthermore, not only the buffer size may be adjusted, but also the particular codec used may be changed at run-time to avoid interruptions and improve the overall QoE, as suggested (but not evaluated) in our first implementation already.

Acknowledgment

We thank IBM Research and Vodafone Foundation Czech Republic for their kind support. Our research was supported by grants FRVS 2621/2012 and SGS11/123/OHK3/2T/13.

REFERENCES

[1] Google Play. (2012, April) Sipdroid. [Online]. Available: https://play.google.com/store/apps/details?id=org.sipdroid.sipua

[2] P. Arlos and M. Fiedler, "Influence of the packet size on the one-way delay on the down-link in 3G networks," in *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*, May 2010, pp. 573 – 578.

[3] T. Hossfeld and A. Binzenhöfer, "Analysis of Skype VoIP Traffic in UMTS: End-to-End QoS and QoE Measurements," *Computer Networks, online http://dx.doi.org/10.1016/j.comnet.2007.10.008*, vol. Vol 52/3 pp 650-666, February 2008.

[4] J. Prokkola, P. Perala, M. Hanski, and E. Piri, "3G/HSPA Performance in Live Networks from the End User Perspective," in *Communications, 2009. ICC '09. IEEE International Conference on*, June 2009, pp. 1 – 6.

[5] L. Sousa, M. Carvalho, E. Rodrigues, L. Sampaio, and F. Cavalcanti, "Quality of service evaluation of VoIP over HSDPA," in *Telecommunications Symposium, 2006 International*, Sept. 2006, pp. 706 – 711.

[6] J. Prokkola, M. Hanski, M. Jurvansuu, and M. Immonen, "Measuring WCDMA and HSDPA Delay Characteristics with QoSMeT," in *Communications, 2007. ICC '07. IEEE International Conference on*, June 2007, pp. 492 – 498.

[7] M. Varela and J. Laulajainen, "QOE-driven mobility management - Integrating the users' quality perception into network-level decision making," in *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on*, Sept. 2011, pp. 19 – 24.

[8] International Telecommunication Union. (1996, August) Recommendation P.800: Methods for subjective determination of transmission quality. [Online]. Available: http://www.itu.int/rec/T-REC-P.800

[9] International Telecommunication Union. (2011, December) Recommendation G.107: The E-model: a computational model for use in transmission planning. [Online]. Available: http://www.itu.int/rec/T-REC-G.107

[10] N. Costa and M. Nunes, "Adaptive Quality of Service in Voice over IP Communications," in *Networking and Services, 2009. ICNS '09. Fifth International Conference on*, April 2009, pp. 19 – 24.

[11] L. Ding and R. Goubran, "Assessment of effects of packet loss on speech quality in VoIP," in *Haptic, Audio and Visual Environments and Their Applications, 2003. HAVE 2003. Proceedings. The 2nd IEEE Internatioal Workshop on*, Sept. 2003, pp. 49 – 54.

[12] L. Ding and R. Goubran, "Speech quality prediction in VoIP using the extended E-model," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 7, Dec. 2003, pp. 3974 – 3978.

[13] K. Patel, S. Anand, and S. Kumar, "A Novel Scalable Architecture for Efficient QoS to Cater IMS Services for Handheld Devices Based on Android," in *Next Generation Mobile Applications, Services and Technologies (NGMAST), 2010 Fourth International Conference on*, July 2010, pp. 106 – 111.

[14] H. Koumaras, N. Zotos, L. Boula, and A. Kourtis, "A QoE-aware IMS infrastrusture for multimedia services," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, Oct. 2011, pp. 1 – 7.

[15] B. Huntgeburth, M. Maruschke, and S. Schumann, "Open-Source Based Prototype for Quality of Service (QoS) Monitoring and Quality of Experience (QoE) Estimation in Telecommunication Environments," in *Next Generation Mobile Applications, Services and Technologies (NGMAST), 2011 5th International Conference on*, Sept. 2011, pp. 161 – 168.

[16] F. Neves, S. Cardeal, S. Soares, P. Assuncao, and F. Tavares, "Quality model for monitoring QoE in VoIP services," in *EUROCON - International Conference on Computer as a Tool (EUROCON), 2011 IEEE*, April 2011, pp. 1 – 4.

[17] A. Arjona, C. Westphal, A. Yla-Jaaski, and M. Kristensson, "Towards High Quality VoIP in 3G Networks - An Empirical Study," in *Telecommunications, 2008. AICT '08. Fourth Advanced International Conference on*, June 2008, pp. 143 – 150.

[18] S. Jelassi and G. Rubino, "A comparison study of automatic speech quality assessors sensitive to packet loss burstiness," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, Jan. 2011, pp. 415 – 420.

[19] S. Paulsen, T. Uhl, and K. Nowicki, "Influence of the jitter buffer on the quality of service VoIP," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, Oct. 2011, pp. 1 – 5.

[20] B. Oklander and M. Sidi, "Jitter Buffer Analysis," in *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on*, Aug. 2008, pp. 1 – 6.

[21] J. Zhou, X. She, and L. Chen, "Source and Channel Coding Adaptation for Optimizing VoIP Quality of Experience in Cellular Systems," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, April 2010, pp. 1 – 6.

[22] S. Jelassi, H. Youssef, and G. Pujolle, "Parametric speech quality models for measuring the perceptual effect of network delay jitter," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, Oct. 2009, pp. 193 – 200.

[23] International Telecommunication Union. (1988, Nov.) Recommendation G.711: Pulse code modulation (PCM) of voice frequencies. [Online]. Available: http://www.itu.int/rec/T-REC-G.711

[24] International Telecommunication Union. (2007, July) Recommendation G.722.2: Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB). [Online]. Available: http://www.itu.int/rec/T-REC-G.722.2

[25] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. (2003, July) RTP: A Transport Protocol for Real-Time Applications. [Online]. Available: http://tools.ietf.org/html/rfc3550

[26] International Telecommunication Union. (2003, May) Recommendation G.114: One-way transmission time. [Online]. Available: http://www.itu.int/rec/T-REC-G.114

[27] L. Cai, X. Shen, J. Pan, and J. Mark, "Performance analysis of TCP-friendly AIMD algorithms for multimedia applications," *Multimedia, IEEE Transactions on*, vol. 7, no. 2, pp. 339 – 355, April 2005.