A QoE Testbed for Socially-Aware Video-Mediated Group Communication

Marwin Schmitt
CWI: Centrum Wiskunde & Informatika
The Netherlands
schmitt@cwi.nl

Simon Gunkel
CWI: Centrum Wiskunde & Informatika
The Netherlands
gunkel@cwi.nl

Pablo Cesar
CWI: Centrum Wiskunde & Informatika
The Netherlands
p.s.cesar@cwi.nl

Peter Hughes
BT Research & Technology
United Kingdom
peter.j.hughes@bt.com

ABSTRACT
Video-Mediated group communication is filtering into everyday use, as commercial products enable people to connect with friends and relatives. Current solutions provide basic support, so that communication can happen, but do they enable conversations? This paper argues that the purpose and the context of the conversation are influential factors that are rarely taken into consideration. The aim should be on the development of underlying mechanisms that can seamless palliate the effects of networking variances (e.g., delays) and optimize media and connection for every single participant. In particular, our interest is on how to improve remote multi-party gatherings by dynamically adjusting network and communication parameters, depending on the ongoing conversation. If we are to provide a software component that can, in real-time, monitor the Quality of Experience (QoE), we would have to carry out extensive experiments under different varying (but controllable) conditions. Unfortunately, there are no tools available that provide us the required fined-grained level of control. This paper reports on our efforts implementing such a testbed. It provides the experiment conductor with the possibility of modifying and monitoring network and media conditions in real-time.

Categories and Subject Descriptors
H.4.3 [**Communications Applications**]: videoconferencing H.5.1 [**Multimedia Information Systems**]: Evaluation/methodology H.5.3 [**Group and Organization Interfaces**]: Evaluation/methodology, Synchronous interaction

General Terms
Experimentation, Human Factors

Keywords
QoE, QoS, videoconferencing, measurements, testbed

1. INTRODUCTION
As video-mediated group communication gradually finds it ways into our everyday life, we need to build systems that support our needs whether we are just casually catching up with family overseas or watching the latest game of our favorite sports team with far away friends. Current systems adapt the media to the network conditions [1] or change the layout to portrait the loudest location (Google+ Hangout).

To build socially-aware systems, we need a better understanding on how asymmetric network conditions, group activities and different roles affect the individual and the overall QoE (e.g., how delay on a single participant affects the overall QoE? Should we provide more bandwidth to active participants over passive participants? Should we use more of the available bandwidth for frame-rate or resolution?)

Previous research investigated dyadic conversations [2], high-end business-oriented solutions [3] and the implication of de-synchronization when watching videos remotely [4]. But they only provide us with an starting point. ITU is starting to look into the direction of QoE assessment of multiparty tele-meetings as well [5]. Still there are no recommendations available and the current knowledge is not sufficient to build systems, which can act upon the influencing factors of QoE.

Such knowledge is obtained through extensive user trials under diverse, but controlled, conditions. Unfortunately, none of the publicly available solutions provide the flexibility and level of control, which is required for extensively investigate the influence of network and media parameters on the QoE. We investigated how such experiments can be done with Google's Hangout but we ran into several problems. The control and manipulation of the technical aspects are only indirectly possible through simulating network conditions. If we are to investigate asymmetric network conditions this requires an extensive infrastructure. Monitoring the experiment sessions becomes also problematic. In standard video-conferencing software, the experiment conductor cannot be hidden, which influences the trial. Solutions for recording the media streams, in the original and degraded version, are either accompanied by quality reduction, which does not allow reasoning about the original perceived quality or require expensive specialized hardware.

This paper tries to fill a current gap: the lack of an adequate testbed for controlled experiments, which allows obtaining conclusive results regarding QoE in video-mediated group communication. It describes our solution the Video-Mediated Communication Testbed (VMC-TB).

2. QoE in Video Mediated Communication
In order to evaluate the QoE in video mediated communication (VMC) we have to address two issues: *what factors* do we look at and *how do we measure* them.

2.1 Factors of QoE in VMC
The model we use is based on the framework by Geerts et al. [6] and the one by Wu et al. [7]. We applied them to VMC and consider in this paper only aspects for controlled experiments. This model, visualized in Figure 1, considers the different factors that will influence QoE in video-mediated communication. QoE,

SAM'13, October 21 2013, Barcelona, Spain

http://dx.doi.org/10.1145/2509916.2509928

as a cognitive process, is located in the center. The distance of the individual factors to the center denotes how strongly they influence the current experience. The model has three dimensions: System, User and Context.

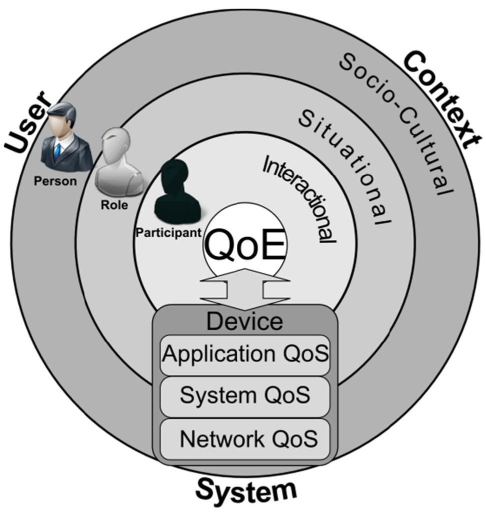

Figure 1 Influence Factor Model for QoE in VMC

The **System** dimension represents the technical aspects. This dimension is modeled as a stack. The lowest layer contains the network aspects, which influence the system layer, which in turn influence the application layer (the user perceptible aspect). At this layer is where QoS and QoE interface, as the user interacts with the system. Real-Time QoS monitoring usually traces network aspects like packet delay, jitter and bandwidth, but users only judge the end result like video artifacts and audio quality. The impact of the network level on the application level is shaped by numerous factors like protocol and codecs. The parameters of QoS impacting QoE in VMC, based on the model for distributed interactive systems by Wu et al. [7], are shown in Figure 2.

QoS stack is embedded in the device and accessed through the system UI, which also directly interacts with the user, in this paper we are focusing on dynamic aspects influenced by the network.

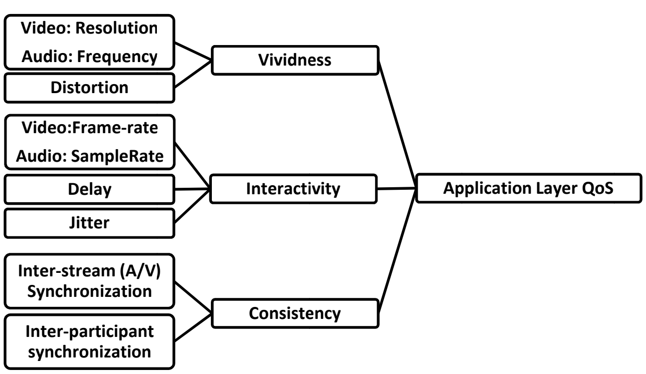

Figure 2 QoS Parameters interfacing QoE (adapted from [7])

The **user** dimension distinguishes between a person and the roles this person takes in video-mediated conversations [6]. On the participant level, we are concerned with experiences that are typical for a general user of a VMC system. On the role level, we want to detail experiences, based on the assumption, that a user adopts a specific role in a specific context. E.g. Given some delay, the experience from the moderator might be different from the one of other participants. Finally, at the person level we consider the individual experience this user has. The relation from more general to very individual experiences is illustrated in Figure 1 by the level of detail the user icon has.

The **context** is created by the interplay of the user with the situation. The user reacts to based on how she/he perceives the situation and in turn shapes it with her/his actions which makes the context inherently unstable [8]. The context can be classified based on three levels [6] [8]: interaction, situational and socio-cultural.

The interactional context covers the interaction of the user with the system and the current task at hand. The interactional context is embedded in the situational context, which concerns the session in terms of activities and participants. The user has certain interests from which she/he forms, given the opportunities of the current situation, her/his current goals. The socio-cultural context, in which these situations take place, deals with aspects like societal conversation etiquette and habits. The socio-cultural layer can be modeled as a reciprocal interaction between social norms and actions people evaluate, plan and carry out.

2.2 Measuring Methodologies

The impact of these factors is modeled through an adaption of the basic process from environmental psychology. In this process environmental influences form cognitive perceptions and lead to behavioral consequences [9]. It is still an ongoing research to determine which cognitive perceptions should be considered for QoE, but it is clear that we have to measure the behavioral consequences. The measurements are usually categorized into subjective and objective methods[7].

Subjective methods are self-reports from the user, giving insights about how the participant perceived the session from his or her own point of view.

Questionnaires are a common methodology for assessing impressions in a quantitative way. They allow processing the feedback with statistical methods.

Interviews are commonly used for gathering qualitative feedback. They provide a descriptive view of the users experience and can provide more detailed insights about it.

Objective methods are based on externally observable and quantifiable behavioral changes.

Task Scores are metrics based on the task the participants have to perform in the experiment. Common metrics are task completion times or successful vs. unsuccessful attempts. Task scores are, in VMC, often only an indirect measure, since many scenarios do not have an inherent quantifiable task.

Speech Patterns are a more direct look at the ongoing interaction. For this a model of turn-taking [10] is applied which quantifies the ongoing interaction in terms of speaking times, length of turns and pauses, simultaneous speech etc. Previous research has found influences on speech patterns from the previously detailed factors: mediating technology [11], roles [12] and context [13].

Physiological measurements are based on biological reactions (e.g. heart rate) which are correlated with experiences.

3. Video-Mediated Conversation-TestBed

In this section we present our developed testbed. We first give a quick overview of the system. Then we explain how the system

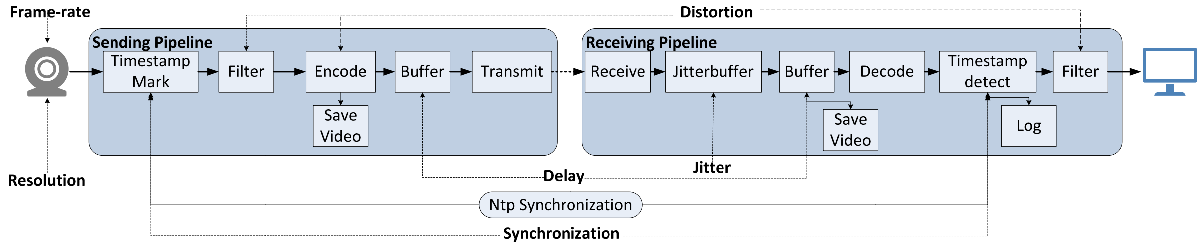

Figure 4 Video Pipelines in VMC-TB

enables us to investigate the factors highlighted in Section 2.1 and how the methodologies, detailed in Section 2.2, are integrated.

3.1 Video Client for Multiparty Conferencing

Our testbed consists of a Video-Client for Multiparty Conferencing, shown in Figure 3, an ObserverControl Client for the experiment conductor and a tool for analyzing experiment sessions (Figure 5).

Figure 3 VMC-TB Client

The clients are full-featured multiparty-video conferencing applications which are directly connected with each other. The system is designed so it runs in a controlled environment. At the moment we transmit data over User Datagram Protocol (UDP). We implemented the media processing pipelines of our testbed using GStreamer[1], a flexible, open-source toolkit with source-filter-sink based architecture. Figure 4 shows a simplified version of a sending and a receiving pipeline for the video stream. Besides the normal elements for capturing, encoding, and transmission, we added elements for monitoring and controlling the network and media parameters (see section 3.2.1). While GStreamer is implemented in C, we implemented the not-so performance critical components in the more lightweight programming language Python. This gives us a flexible platform, which is easily extensible and customizable .

The experiment conductor (using the ObserverControl Client), is usually not shown to the other participants, not to influence the trial, but can dynamically join the conversation, if necessary, to give feedback or additional instructions. Furthermore, the different steps of the experiment can be scripted based on the status of the system. E.g. to automatically show a questionnaire after a task is finished, set new conditions after all questionnaires are filled out and so forth. Each individual step is logged and available for data analysis.

[1] www.gstreamer.org

3.2 Support for QoE Factors

In the following, we explain how the main QoE factors are supported in our testbed.

3.2.1 QoS Parameters

In section 2.1 we identified some QoS parameters that will have an impact on the QoE. To investigate the effects of these parameters, we need to be able to control and monitor them. For monitoring the visual (for audio respectively acoustical) aspects, we record the transmitted streams on the sender and receiver side. To keep track of the temporal aspects, we synchronize the clocks of our clients via the Network Time Protocol (NTP) and log the delay of every frame. For this we directly insert a barcode into the video, which we crop-out at the receiving side before presenting the video to the user (compare Figure 3 and Figure 5). By directly inserting the timestamps into the video we measure the delay of the whole processing pipeline, instead of only of the network delay. For the complete “mouth-to-ear” delay we need to also consider the delay of capturing and rendering equipment, which can be assumed to be static and can be measured using external tools [14].

The parameters Resolution/Frequency and Frame-rate/Sample-rate can be manipulated directly at the corresponding capturing elements (with respect to the capabilities of the devices). For the other parameter, we use the following:

- **Distortion**: We can control distortion by inserting available filters from GStreamer (e.g. blur) or changing the codec settings. The easy extensible plugin architecture of GStreamer makes it easy to develop and integrate custom, more complex distortion patterns.
- **Delay**: The minimum delay our system achieves, in the ideal conditions of our local network, is in average 70ms with a 25ms standard deviation. We can add delay by increasing minimum amount of data hold in the buffers on the sending and the receiving side.
- **Jitter**: We keep the network delay constant by employing a jitterbuffer. We can add jitter by adjusting the buffer on the receiving side.
- **Interstream (Audio/Video) Synchronization**: We can achieve audio/video (de)synchronization by manipulating the delay buffers in audio and video streams separately.
- **Inter-participant Synchronization**: Since there is a separate pipeline for every participant we can achieve basic (de)synchronization by setting different delays for each participant. Since we have synchronized clocks and the capturing timestamps more complex synchronization algorithm can be built on top of this.

All parameters can be modified in real-time at a running system, which is required, since modern networks have typically fluctuating performances during one session [15].

3.2.2 *User*

We achieve the monitoring of each individual participant by recording the received and sent video at each client. This allows an investigation from the perspective of each user.

The roles arise from the experiment design. Forcing users into formal roles can be achieved by assigning them to the participants in the experiment (e.g. discussion moderator) or creating scenarios with roles present (e.g. tele teaching with teacher and student). We support the experiment conductor in this task, by allowing him to label each participant with an individual a role. This label can be used to asymmetrically manipulate the aforementioned parameters, execute specific behavior for the activity and it is available as metadata for the data analysis after the experiment.

Further insights into the user's personal traits can be gained by integrating corresponding questions into the questionnaires before or after the experiments. Biases, attitudes and expectations with the experiment scenario and VMC systems in general should be considered.

3.2.3 *Context*

The interaction context is shaped by the user, when interacting with the system under a given task [6]. The User Interface of the VMC-TB Client is designed to support easy customization. It can be easily adapted for a specific experiment with the Glade GUI-Builder[2].

For example, the client shown in Figure 3 is designed for an experiment that investigates the effect of delay in semi-structured group discussions with 5 participants. The client shows a small version of the own video in the upper left corner, a task specific pane below it and the other 4 participants in a square layout. In the task pane, we implemented a shared view of the questions participants had to discuss and had to select an answer from. In this experiment we decided to make a static layout, which shows all participants in the same size, as we focused on delay, and wanted to keep the influence from layout as constant as possible. The UI is rather simple, based on the experience in UI design, that participants give more feedback on systems which are recognizable as prototypes [16]. Further than the layout and task integration, the local context is shaped by the specific experiment design. The situational context is in controlled experiments always imposed ("participating in an experiment") and the socio-cultural context by inviting the appropriate participant to the study. Further insights into the socio-cultural background of the participants can be gained through questionnaires before or after the experiment session. Questionnaires to assess the socio-cultural background should thus investigate knowledge, experiences and plans of activities similar to the experiment activity and VMC in general.

3.3 Assessing Feedback

Questionnaires

VMC-TB integrates functionality so questionnaires can be administered. The questionnaires can be easily defined over a simple document format and displayed to the participants throughout the experiment. The experiment conductor receives the results from the questionnaires directly after they have been filled out. The integration of the experiments into the testbed has many advantages. The questionnaire can be made an intrinsic part of the experiment, so completion of the questionnaire can trigger the next step of the experiment. Furthermore it is easy to dynamically integrate aspects of the session at hand, e.g. questions about specific participants or based on the completion of the task.

Task Scores

The integration of tasks makes it possible to make an automatized scoring of the outcome. In the example sketched in Section 3.2.3 the answers of the decision making task was transmitted to the observer during the experiment. Additionally, other more traditional measures such as completion time are easily obtained, as the testbed logs all experiment steps.

Speech Patterns

The transmitted media is saved on the sending and on the receiving side. This enables us to analyze the conversation after the experiment from each participant's perspective. We created a tool for viewing an experiment session and analyzing speech patterns (see Figure 5). The tool shows for each participant a timeline with blocks when there is audio activity. These are usually speech (shown in light blue), and can be later tagged by the analyst by different categories, e.g. laughter shown in green or non-verbal utterance (e.g. "uh", "hmm") shown in orange.

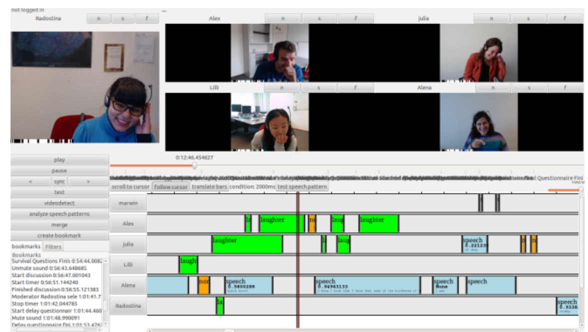

Figure 5 Speech Pattern Analysis Tool

Interviews and **physiological measurements**

Our tool does not specially supported interviews or physiological measurements. Interviews are indirectly aided, as our system makes it easy for the experiment conductor to have a first look at the questionnaire data during or directly at the end of the trial. This makes it possible to focus in the debriefing on areas where additional information might be needed for the interpretation of the data.

4. DISCUSSION

We reported in this paper a testbed for conducting experimental research. We showed that the software permits us to control and directly manipulate a number of application level QoS parameters. The monitoring capabilities allow us, to analyze in detail the conducted experiments.

The tool allows us to consider different scenarios and activities. By integrating the task into our system, and monitor it, we can better understand what kind of interaction patterns arise and how they change under the influence of our QoS parameters. The detailed monitoring of each participant, allows us to investigate, whether specific effects are dependent on the role of the user. With the analysis of speech pattern, we have a promising tool, which provides objective data that complements the subjective user feedback. The recording of the media streams in original and degraded quality, allow us to investigate whether current objective full-reference QoE metrics, correlate with the subjective feedback. As a first step we started to investigate the effect of delay on semi-structured group discussion with 5 participants. We used a decision making task, similar to the survival scenario [5] and assigned one of the participants to be the moderator. As a starting

[2] glade.gnome.org

point, we conducted a study with around 50 participants, setting symmetric conditions for all participants. In the next step, we will asymmetrically modify the delay conditions, to gain insight how this impacts the individual vs. the overall QoE of the group. This will give us insight what are appropriate inter-participant synchronization schemes for group discussions.

Further, we want to investigate the effect of the parameters video and audio quality and common VMC activities like sharing media. The testbed is a first step towards our final goal: detailing the effect from different factors (QoS, context, roles) on the individual and overall QoE in video-mediated group communication.

5. REFERENCES

[1] L. De Cicco, S. Mascolo, and V. Palmisano, "Skype video responsiveness to bandwidth variations," in *Proc. of NOSSDAV'08*, 2008, pp. 81–86.

[2] J. Tam, E. Carter, S. Kiesler, and J. Hodgins, "Video increases the perception of naturalness during remote interactions with latency," in *Proc. of CHI'12*, New York, NY, USA, 2012, pp. 2045–2050.

[3] E. Geelhoed, A. Parker, D. J. Williams, and M. Groen, "Effects of Latency on Telepresence," HP labs technical report: HPL-2009-120 http://www. hpl. hp. com/techreports/2009/HPL-2009-120. html, 2009.

[4] D. Geerts, I. Vaishnavi, R. Mekuria, O. Van Deventer, and P. Cesar, "Are we in sync?: synchronization requirements for watching online video together.," in *Proc. of CHI'11*, 2011, pp. 311–314.

[5] P. 130. ITU-T RECOMMENDATION, "ITU-P.1301 - Subjective quality evaluation of audio and audiovisual multiparty telemeetings." 27-Feb-2013.

[6] D. Geerts, K. De Moor, I. Ketyko, A. Jacobs, J. Van den Bergh, W. Joseph, L. Martens, and L. De Marez, "Linking an integrated framework with appropriate methods for measuring QoE," in *QoMEX'10*, 2010, pp. 158–163.

[7] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, and Z. Yang, "Quality of experience in distributed interactive multimedia environments: toward a theoretical framework," in *Proc. of ACM MM'09*, New York, NY, USA, 2009, pp. 481–490.

[8] G. Mantovani, "Social Context in HCl: A New Framework for Mental Models, Cooperation, and Communication," *Cogn. Sci.*, vol. 20, no. 2, pp. 237–269, 1996.

[9] A. M. and J. A. Russell, "An Approach to Environmental Psychology," Apr. 1980.

[10] H. Sacks, E. Schegloff, and G. Jefferson, "A Simplest Systematics for the Organization of Turn-Taking for Conversation," *Language*, vol. 50, no. 4, pp. 696–735, 1974.

[11] L. Ten Bosch, N. Oostdijk, and J. de Ruiter, "Durational aspects of turn-taking in spontaneous face-to-face and telephone dialogues," in *Text, Speech and Dialogue*, 2004, pp. 563–570.

[12] W. Dong, B. Lepri, A. Cappelletti, A. S. Pentland, F. Pianesi, and M. Zancanaro, "Using the influence model to recognize functional roles in meetings," in *Proc. of ICMI'07*, New York, NY, USA, 2007, pp. 271–278.

[13] M. Jochemsen, "The Effect of Context on Turn Taking in Human to Human Conversation," presented at the TSConIT'11, 2011.

[14] J. Jansen and D. C. A. Bulterman, "User-Centric Video Delay Measurements," presented at the NOSSDAV, 2013.

[15] I. T. S. Sector, "ITU-T Recommendation P. 880," *Contin. Eval. Time-Varying Speech Qual.*, 2004.

[16] A. Black, "Visible planning on paper and on screen: The impact of working medium on decision-making by novice graphic designers," *Behav. Inf. Technol.*, vol. 9, no. 4, pp. 283–296, 1990.