Summary Review Documentation for

“Video Telephony for End-consumers: Measurement Study of Google+, iChat, and Skype”

Authors: Y. Xu, C. Yu, J. Li, Y. Liu

Reviewer #1

Summary: This paper is presenting a measurement study of three video telephony systems: iChat, Google+ Hangout, and Skype. The authors combine a very interesting number of active and passive experiments to understand the structure of the three services and quantify their performance.

Strengths:

- Very nice measurement paper with a combination of active and passive measurement experiments
- Measurement methodology allows visibility into the structure of each service and the mechanisms employed to enable real time, high rate communication
- Results on the performance of Google+ Hangout, iChat, and Skype, and identification of the mechanisms that result in better performance and under what conditions

Weaknesses: The part of the paper that deals with the discovery of the structure of the 3 services is not methodical right now - one would not be able to reproduce the authors findings just by reading the text (some kind of pseudo-algorithm would be very helpful for reproducibility)

Comments to authors: This is one of the best measurement papers I have read in a while. The authors very carefully design a number of active experiments and analyze the collected packet traces with the goal to essentially reverse engineer the infrastructure behind Google+ Hangout, iChat, and Skype, as well as the mechanisms that they have implemented for the support of high rate, real time video telephony. Interestingly, the authors demonstrate that the 3 services utilize not only different infrastructure, but also different mechanisms to deal with loss. They are further able to quantify the performance of the 3 services under a number of networking conditions. Truly exceptional in my mind.

Having said that I think a few points could possibly be improved: 1) the English:-) having the paper read by a native English speaker would help 2) Table 6 presents the payload of Google+ Hangout that allows them to make the observation that Google+ Hangout employees multi-layer coding. Is there a way you could verify this behavior in a quantifiable fashion across the entire packet trace instead of relying on visual inspection of a snippet? 3) beyond pure observation is there a way you could define pseudo-algorithms that capture the behavior that you extract from the experiments that allowed you to identify the structure of the supporting infrastructure (client/server, p2p, hybrid), and the support of multi-layer coding?

Such “pseudo-algorithms” would allow for reproducibility of your results, which right now may be hard based on the current text. 4) Is there some logic one could extract from Figure 6? How does the system determine after how much time to perform the next retransmission? Have you figured out the rule based on your experiments? 5) In your conclusions of Section 8, I would also add a point on the effectiveness of FEC over high RTT links.

All in all, I really enjoyed reading the paper. I would really like to see a version where reproducibility is improved due to a few more details on how one could “codify” the analysis you performed.

Reviewer #2

Summary: This paper reverse engineers the video conferencing strategies of Google+, iChat, and Skype. It reports on their overall architecture (server-centric vs. p2p), encoding strategies, delay, and mechanisms to adapt to network conditions. The experiments are detailed and appear carefully done.

Strengths: There is a wealth of detailed information about in this paper about the three systems. Several communities (e.g., codec designers and future conferencing system designers) will find this information useful.

Weaknesses: The paper essentially captures the design decisions of three engineering teams which probably faced very different requirements (e.g., for Google engineers using the infrastructure was no big deal but Apple engineers didn’t have that luxury). A more meaningful evaluation would have considered those constraints as well.

Somewhat related, while there are some hints, there is little direct analysis that suggests, “this is the right way to build a video conference system,” or “one of the systems could be significantly improved only if they did X differently.”

Comments to authors: I enjoyed reading this paper and learned a fair bit about choices made by three systems. The diversity of those choices is fascinating (which as I mentioned above likely stem largely from the constraints that the engineering teams had to work with).

I found the conclusion section to be rather weak and some of the listed findings not really supported in the paper. For example, the third finding is that “compared to multi-version coding, scalable video codign can simultaneously address user access heterogeneity and network bandwidth varitions.” I didn’t think that you showed this at all. There are experiments with Google, but they show that Google uses SVC, but that is about it.

You end with "Our study demonstrated the benefits of jointly designing video generation, adaptation, and distribution to deliver high-quality video telephony" Where and how exactly you showed that? This goes back to my point above in the weakness section. Your work reports on current decisions in the systems, and do not inform me about good, alternate designs.

I'd suggest that you either provide direct evidence on your listed findings or modify your findings to those that directly supported by experiments.

How did you decide that Skype servers are in New York? If they were in New York and if routing was not screwed up, you shouldn't have seen 26 ms RTT from New York.

I am amazed that Skype goes all the way down to 5 kbps. How does the video look at that point?

Is the mapping from rate to video parameters (resolution, fps, and quantization) 1:1? I am curious if the video rate needs to be reduced, how is it decided which parameter to tweak.

The tricks to measure voice and video delay are clever!

The differences in the systems for one-way delay are interesting. Clearly, the architectural differences do not explain those differences. Some of it must be how voice and video are processed. So, a couple of questions:

- what is responsible for these differences?
- did you measure the CPU usage of these clients? do different clients differ in their aggressiveness towards consuming my CPU, which then impacts one-way delay?

Why didn't you report on one-way delay for iChat?

In 7.2, you report that the Skype relay server is doing something funky with FEC reprocessing? Is that contrary to your results that the server doesn't do any transcoding? Or, you don't consider FEC reprocessing as transcoding?

Reviewer #3

Summary: This paper uncovers the underlying technologies of video telephony used in Google+, iChat and Skype. It also compares the three systems in terms of latency and loss and offers insights into their performance differences.

Strengths:

- Very thorough study on the key technologies employed by three popular video telephony systems.
- Comparison results can guide the optimization of these systems.

Weaknesses: The paper puts too much emphasis on the technological differences among the three systems, which I think is less important. Instead, it should shift the focus onto their performance differences and the reasons behind.

Comments to authors: This paper presents a detailed study on the three widely-used video telephony systems on the market, namely Google+, iChat, and Skype. It examines these systems from several key aspects that are crucial to their performance, including

topology (p2p vs. client-server), video encoding (single- vs. multi-version), and loss recovery (FEC vs. retransmission). Moreover, it compares delay and loss differences among these systems and stems the reasons behind the observed differences.

Overall, it is clearly written and easy to understand. The topic is certainly of interest to the IMC community. I learned a lot of interesting facts from it.

My major comment relates to the organization of the paper. The first half of the paper (up to sec 5) is entirely about the various technologies used by these systems. Although the experimental methodology is pretty cute and the results are interesting, you didn't explain why one would care about them besides out of "pure curiosity". It wasn't until sec 6 where I started to see the connections between the technologies and user experiences. You should have mentioned this much earlier in the introduction.

The key findings of the study are listed in the conclusion, which again is too late. It is also a bit difficult to link each finding to the relevant results in the earlier sections.

Sec 6.2 and 6.3, you show the one-way and round-trip delays. Could you further break them down into network, sender and receiver parts? This would reveal the dominating factor in the end-to-end delays.

Fig 5 is fascinating. You attributed Skype's high delay (variations) to the public Internet. I thought you could confirm this by conducting more detailed experiments.

Reviewer #4

Summary: The paper uses some fairly interesting black box analysis techniques to do a detailed comparative study of three different internet video chat systems, iChat, Skype video and G+. The study is quite comprehensive, covering techniques for failure recovery, video/audio stream transmission encoding and relay paths, and fault recovery schemes. The results are also interesting, and shed significant insight on how/why different systems behave under different conditions, and how different techniques led to those performance results.

Strengths: The study is comprehensive, and many of the techniques are quite inventive (and often effort intensive). The approaches with video and audio delay capture show considerable thought went into the design of the measurement experiments.

The results are surprising, and very informative on a variety of fronts, including the effectiveness of different fault-recovery and reliability techniques, as well as the impact by different approaches for traffic forwarding on e2e user observed performance.

Weaknesses: Not too many weaknesses, some minor gripes:

- there are some statements in section 4 that could use more explanation. Mentions of tcp in rare cases for example, naturally raise questions of "what rare cases"?
- section 4.5 is fairly useless, since the results basically show that the geolocation db is wrong, but yet no actual results are definitively shown.
- some of the text form very long wordy paragraphs. more breaks into distinctive concise paragraphs would make the pa-

per much more readable, e.g. 6.1.2 is 1 para, first para of 7.3 is also too long

- some of the results could be presented more clearly in chart form rather than tables, e.g. table 16.

Comments to authors: This is a rather surprising paper. My expectation were fairly low, given the amount of work already in this space, but I was very pleasantly surprised by the inventiveness and the detailed black-box measurements that the authors designed and carried out. Some of the effort given was quite impressive (and sounded quite painstaking).

The results cover quite a wide range of topics, and actually manage to shed surprising light on most of them. Aside from the geolocation results which were not useful, there were nice empirical take-aways from the encoding section, the arch section, the recovery section, etc. Overall, lots of nice conclusions drawn and justified.

Reviewer #5

Summary: This paper presents a characterization of three popular video conferencing services: Google+, Skype, and iChat in terms of their application architectures, server placement strategies, and approaches to deal with poor network conditions.

Strengths: Interesting, new problem that I have not seen much related work on. Some of the measurement methodology is pretty cute (especially in Section 6)

Weaknesses: Not sure the paper has a significant new contribution in terms of measurement methodology or insights beyond the reverse engineering exercise.

The writing can be significantly improved to highlight the key contributions and findings and takeaways/implications of the findings.

Comments to authors: I believe this work has a lot of potential, but I could not get very excited by the current presentation as I struggled to identify new contributions or insights from the paper.

Also, what are the implications of your measurements in terms of improving the quality of these systems? – Is one application architecture or design choice significantly better than others? I understand your goal to not do a head-to-head comparison between the systems, but I think that avoiding an architectural comparison is a lost opportunity on providing new insights to the community!

I would have liked to see some results on comparing the performance of these systems as a function of the number of participants or the number of simultaneous speakers, premium service etc, to see their robustness to such effects. e.g., does the application architecture choice change depending on a two-party vs a multi-party call? Also, Skype offers some "premium" service, does the quality for premium customers significantly differ from the vanilla service?

Response from the Authors

Most of the comment issues are related to the writing. We took a lot of effort to improve this, like listing our experiment contents and their importance in the beginning, describing our experimental methodology more clearly, better connecting experimental results with their implications, and listing our key findings to help build a good conferencing system in the conclusion, etc.

We did more experiments to verify our findings. To better geolocate Skype's server, we did traceroute experiments and additional ping experiments. We tried to block UDP to observe whether these systems can switch to TCP for transmission. To verify the delay performances of these systems in the Internet, we did one more measurement of round-trip delay between New York and Singapore. We also wrote a script across the entire packet trace to verify that Google+ employs multi-layered coding.

Answers to specific reviewers: R1: We can guess Google+'s retransmission rule from Figure 6. Compared to the rule of always waiting for about 70ms before next transmission in the uplink side, Google+ uses a more aggressive batch retransmission in the downlink side. Maybe this design decision is based on the fact that users downlink bandwidth is normally larger than the uplink one. R2: When the rate of Skype goes down to 5 kbps, it shows the lowest resolution (160*120) and the pictures of Skype only change once every several seconds. And we don't consider FEC reprocessing as video transcoding. R3: In our one-way delay measurement, network delay is very small, video encoding and decoding delay are the dominating factors. When running in the Internet, since delay in video encoding and decoding parts are considered to be constant, network delay part is varying for the reason that these systems need to do some processing to deal with unknown packet losses. Thus, we further do the measurement of round-trip delay to show the delay performances of these systems in the Internet. Unfortunately, we don't know how to isolate the processing delays incurred on the sender and receiver side. R4: We mention that the geolocation db is wrong in order to show the reason why we couldn't precisely geolocate servers of these conferencing systems. R5: Skype's group video call is its "premium" service. Only premium users can initiate a multi-party conference call. For the two-party video call, "premium" service and vanilla service are the same. In this paper, P2P means that users in the same conference forward video/audio among themselves, and no dedicated server is involved.