Measurement study of multi-party video conferencing

Lu, Y; Zhao, Q; Kuipers, FA; Van Mieghem, PFA

Publication date
2010

Document Version
Final published version

Published in
Proceedings of ifip networking 2010

Measurement Study of Multi-party Video Conferencing

Yue Lu, Yong Zhao, Fernando Kuipers, and Piet Van Mieghem

Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands
{Y.Lu, F.A.Kuipers, P.F.A.VanMieghem}@tudelft.nl

Abstract. More and more free multi-party video conferencing applications are readily available over the Internet and both Server-to-Client (S/C) or Peer-to-Peer (P2P) technologies are used. Investigating their mechanisms, analyzing their system performance, and measuring their quality are important objectives for researchers, developers and end users. In this paper, we take four representative video conferencing applications and reveal their characteristics and different aspects of Quality of Experience. Based on our observations and analysis, we recommend to incorporate the following aspects when designing video conferencing applications: 1) Traffic load control/balancing algorithms to better use the limited bandwidth resources and to have a stable conversation; 2) Re-encode streams to limit the overall traffic.

This work is, to our knowledge, the first measurement work to study and compare mechanisms and performance of existing *free multi-party* video conferencing systems.

1 Introduction

The demand for video conferencing (VC) via the Internet is growing fast. VC services are provided in two different ways: (1) either utilizing a high-quality VC room system with professional equipment and dedicated bandwidth or (2) implementing a VC application on personal computers. The first category can guarantee quality, but it is costly and limited to a fixed location, while the second category is often free of charge and easy to install and use, although the quality cannot be guaranteed.

In this paper, we focus on studying *free* applications that provide multi-party (≥ 3 users) VC on the Internet, and focus on the following questions:

- *How do multi-party VC applications work?*
- *How much resources do they need?*
- *What is the Quality of Experience (QoE)?*
- *What is the bottleneck in providing multi-party VC over the Internet?*
- *Which technology and architecture offer the best QoE?*

In order to answer these questions we have surveyed existing popular VC applications and among them chose and measured four representative applications to investigate, namely *Mebeam*[1], *Qnext*[2], *Vsee*[3], and *Nefsis*[4].

The remainder of this paper is organized as follows. Section 2 presents related work. In Section 3, eighteen popular VC applications will be introduced and classified. Section 4 describes our QoE measurement scenario. Sections 5 and 6 will show the measurement results obtained. Finally, we conclude in Section 7.

2 Related work

Most research focuses on designing the network architectures, mechanisms and streaming technologies for VC. In this section we only discuss the work on studying and comparing the mechanisms and performance of streaming applications.

Skype supports multi-party audio conferencing and 2-party video chat. Baset and Schulzrinne [1] analyzed key Skype functions such as login, call establishment, media transfer and audio conferencing and showed that Skype uses a centralized P2P network to support audio conferencing service. Cicco *et al.* [2] measured Skype video responsiveness to bandwidth variations. Their results indicated that Skype video calls require a minimum of 40 kbps available bandwidth to start and are able to use as much as 450 kbps. A video flow is made elastic through congestion control and an adaptive codec within that bandwidth interval.

Microsoft Office Live Meeting (Professional User License) uses a S/C architecture and has the ability to schedule and manage meetings with up to 1,250 participants. However, only few participants can be presenters who can upload their videos and the others are non-active attendees.

Spiers and Ventura [3] implemented IP multimedia subsystem (IMS)-based VC systems with two different architectures, S/C and P2P, and measured their signaling and data traffic overhead. The results show that S/C offers better network control together with a reduction in signaling and media overhead, whereas P2P allows flexibility, but at the expense of higher overhead.

Silver [4] discussed that applications built on top of web browsers dominate the world of Internet applications today, but are fundamentally flawed. The problems listed include delays and discontinuities, confusion and errors, clumsy interfacing and limited funtionality.

Trueb and Lammers [5] analyzed the codec performance and security in VC. They tested High Definition (HD) VC and Standard Definition (SD) VC traffic characteristics and their corresponding video quality. In their results, HD provides a better video quality at good and acceptable network conditions, while in poor network conditions HD and SD have similar performance.

[1] http://www.mebeam.com/
[2] http://www.qnext.com/
[3] http://www.vsee.com/
[4] http://www.nefsis.com/leads/free-trial.aspx

Few articles compare the different types of existing free multi-party VC systems or measure their QoE. In this paper, our aim is to provide such a comparison.

3 Survey

We have considered eighteen VC applications, for which we list the maximum frame rate they can support (the best video quality they can provide), the maximum number of simultaneous conference participants, and the category (S/C or P2P) they belong to in Table 1.

Table 1. Popular video conferencing applications.

	Max. frame rate (frames/second)	Max. # of simultaneous video participants	S/C or P2P
Eedo WebClass		6	web-based S/C
IOMeeting	30	10	web-based S/C
EarthLink	30	24	S/C
VideoLive	30	6	web-based S/C
Himeeting	17	20	S/C
VidSoft	30	10	S/C
MegaMeeting	30	16	web-based S/C
Smartmeeting	15	4	S/C
Webconference	15	10	web-based S/C
Mebeam		16	web-based S/C
Confest	30	15	S/C
CloudMeeting	30	6	S/C
Linktivity WebDemo	30	6	web-based S/C
WebEx	30	6	web-based S/C
Nefsis	30	10	S/C
Lava-Lava	15	5	decentralized P2P
Qnext		4	centralized P2P
Vsee	30	8	decentralized P2P

Even though there exist many free VC applications, many of them turn out to be instable once installed. From Table 1, we observe that the maximum frame rate is 30 frames/s which corresponds to regular TV quality. All applications support only a very limited number of participants and the applications that support more than 10 simultaneous participants all use a centralized S/C network structure.

Many other popular online chatting applications (like Skype, MSN, Yahoo messenger, Google talk, etc.) only support multi-party audio conference and 2-party video conference, and therefore are not considered here.

4 Experiments Set-up

We have chosen four representative applications to study:

- *Mebeam*: web-browser based S/C with a single server center.
- *Qnext (version 4.0.0.46)*: centralized P2P. The node which hosts the meeting is the super node.
- *Vsee (version 9.0.0.612)*: decentralized full-mesh P2P.
- *Nefsis (free trial version)*: S/C, network of distributed computers as servers.

We have chosen these four applications because they each represent one of the four architectures under which all eighteen applications in Table 1 can be classified.

We have performed two types of experiments: (1) local lab experiments, composed of standard personal computers participating in a local video conference, in order to investigate the login and call establishment process, as well as the protocol and packet distribution of the four VC applications; (2) global experiments, to learn more about the network topology, traffic load and QoE, when a more realistic international video conference is carried out.

The global measurements were conducted during weekdays of May, 2009, under similar and stable conditions[5]:

- Client 1: 145.94.40.113; TUDelft, the Netherlands; 10/100 FastEthernet;
  Client 2: 131.180.41.29; Delft, the Netherlands; 10/100 FastEthernet;
  Client 3: 159.226.43.49; Beijing, China; 10/100 FastEthernet;
  Client 4: 124.228.71.177; Hengyang, China; ADSL 1Mbit/s.
- Client 1 always launches the video conference (as the host);
- Clients 1, 3 and 4 are behind a NAT.

To retrieve results, we used the following applications at each participant:

- *Jperf* to monitor the end-to-end available bandwidth during the whole process of each experiment. We observed that usually the network is quite stable and that the available end-to-end bandwidth is large enough for different applications and different participants.
- *e2eSoftVcam* to stream a stored video via a virtual camera at each VC participant. Each virtual camera is broadcasting in a loop a "News" video (.avi file) with a bit rate of 910 Kbit/s, frame rate of 25 frames/s and size 480x270;
- *Camtasia Studio 6*. Because all applications use an embedded media player to display the Webcamera streaming content, we have to use a screen recorder to capture the streaming content. The best software available to us was *Camtasia*, which could only capture at 10 frames/s. In order to have a fair comparison of the original video to the received video, we captured not only

[5] We have repeated the measurements in July, 2009 and obtained similar results to those obtained in May 2009.

the streaming videos from all participants, but also the original streaming video from the local virtual camera[6].
– *Wireshark* to collect the total traffic at each participant.

5 Measurement results

5.1 Login and Call establishment process

Mebeam: We open the *Mebeam* official website to build a web video-chat room and all participants enter the room. The traces collected with *Wireshark* revealed that two computers located in the US with IP addresses 66.63.191.202 (Login Server) and 66.63.191.211 (Conference Server) are the servers of *Mebeam*. Each client first sends a request to the login server, and after getting a response sets up a connection with the single conferencing server center. When the conference host leaves from the conference room, the meeting can still continue. *Mebeam* uses TCP to transfer the signals, and RTMP[7] to transfer video and audio data.

Qnext: The data captured by *Wireshark* reveals two login severs located in the US. Each client first sends packets to the login servers to join the network. After getting a response, they use SSLv3 to set up a connection with the login servers. In the call establishment process, each client communicates encrypted handshake messages with 3 signaling servers located in the US and Romania and then uses SSLv3 to set up a connection between the client and the signaling server. When client A invites another client B to have a video conference and client B accepts A's request, they use UDP to transfer media data between each other. In a conference, there is only one host and other clients can only communicate with the host. The host is the super node in the network. When the host leaves the meeting, the meeting will end. If another node leaves, the meeting will not be affected. *Qnext* uses TCP for signaling and UDP for video communication among participants.

Vsee: Each client uses UDP and TCP to communicate with the web servers in login process. In the call establishment process, after receiving the invitation of the host, each client uses[8] T.38 to communicate with each other. *Vsee* has many web servers: during our experiment, one in the Netherlands, one in Canada, and 7 located in the US. *Vsee* has a full-meshed P2P topology for video delivery. However, only the host can invite other clients to participant in the conference. When the host leaves the meeting, the meeting cannot continue. Other peers can

[6] We assess the video quality using the full reference batch video quality metric (bVQM) which computes the quality difference of two videos. Capturing at 10 frames/s a video with frame rate of 25 frames/s may lead to a different averaged bVQM score. However, because the video used has a stable content (there are only small changes in the person profile and background), we do not expect a large deviation in bVQM score with that of the 25 frames/s video. The results are accurate for 10 frames/s videos.

[7] Real-Time Messaging Protocol (RTMP) is a protocol for streaming audio, video and data over the Internet, between a Flash player and a server.

[8] T.38 is an ITU recommendation for fax transmission over IP networks in real-time.

leave without disrupting the meeting. *Vsee* is a video-conferencing and real-time collaboration service. The communication among users is usually of the P2P type using UDP, with automatic tunneling through a relay if a direct connection is not available.

Nefsis: In the login process, the clients first use TCP and HTTP to connect to the Virtual Conference Servers (with IP addresses 128.121.149.212 in the US and 118.100.76.89 in Malaysia) and receive information about 5 other access points from the Virtual Conference Servers. These 5 access points are also the data server centers owned by *Nefsis*, and they are located in the Netherlands (Rotterdam and Amsterdam), in the UK, India, Australia, and Singapore. Afterwards, the clients choose some access points to set up connections via TCP. After entering the conference room, each client communicates with each other through the access point when firewalls/NAT are present at clients, otherwise clients can set-up an end-to-end connection to communicate with each other directly. *Nefsis* uses TCP for signaling, and uses UDP to deliver streaming data.

5.2 Packet size distribution and traffic load

To differentiate between non-data packets, video and audio packets, we performed three local experiments for each application. The first experiment uses two computers with cameras and microphones to have a video conference. In the second experiment, two computers are only equipped with microphones, but without cameras (no video packets will be received). In the third experiment, two computers set-up a connection, both without microphones and cameras (so only non-data packets will be exchanged).

Based on *Wireshark* traces, we could distill for each VC application the packet size range as shown in Table 2:

Table 2. The packet size distribution of *Mebeam*, *Qnext*, *Vsee* and *Nefsis*.

Packet size	$Mebeam$	$Qnext$	$Vsee$	$Nefsis$
Audio packet	$>$ 50 bytes	72 bytes	100 $\sim$ 200 bytes	100 $\sim$ 200 bytes
Video packet	$>$ 200 bytes	50 $\sim$ 1100 bytes	500 $\sim$ 600 bytes	1000 $\sim$ 1600 bytes
Signaling packet	50 $\sim$ 200 bytes	50 $\sim$ 400 bytes	50 $\sim$ 100 bytes	50 $\sim$ 100 bytes

Other interesting observations are: 1) If the person profile or background images in the camera change/move acutely, a traffic peak is observed in our traces. 2) The traffic does not necessarily increase as more users join the conference. Fig. 1 shows the change of the average traffic load at each user when a new participant joins the conference[9]. The decreasing slope after 3 users indicates that *Mebeam*, *Qnext* and *Vsee* all re-encode the videos in order to reduce/control the overall traffic load in the system. We can see from Fig. 1 that only the traffic load

[9] We captured the packets after the meeting was set up and became stable.

at *Nefsis* clients does not decrease when the number of video conferencing participants reaches to 4. Therefore, we introduced more participants into the video conference for *Nefsis*, and we found that the traffic at each *Nefsis* user starts to decrease at 5 participants. Hence, we believe that in order to support more simultaneous conference participants, we have to upper-bound the overall traffic and degrade the streaming rate of each participant using adaptive re-encoding technology.

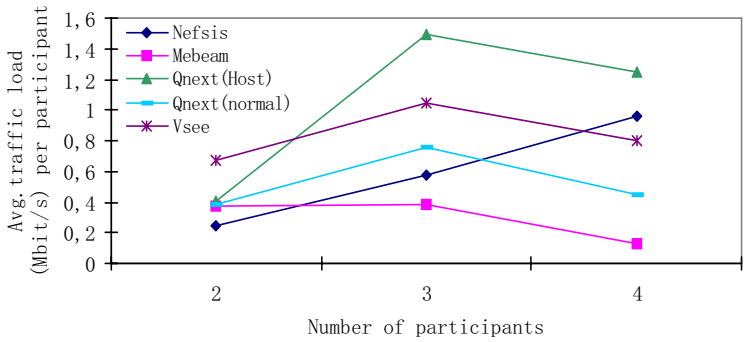


Fig. 1. The average traffic load at an end-user when the number of conference participants increases from 2 to 4 (*Qnext* is limited to 4 participants).

Fig. 1 illustrates that, compared with the traffic generated by *Nefsis* which uses the same coding technology and the same frame rate on the same video, *Qnext* and *Vsee* generate most traffic, especially the host client of *Qnext*. This is because *Qnext* and *Vsee* use P2P architectures where the signaling traffic overhead is much more than the traffic generated by a S/C network with the same number of participants. The host client (super node) of *Qnext* generates 3 times more traffic than other normal clients. Hence, for this architecture, a super-node selection policy is recommended to choose a suitable peer (with more resources, for example) as the super node.

Fig. 1 also shows that *Mebeam* generates least traffic. Considering that the overall traffic load, which can be supported in a VC network, has an upperbound due to the limited users' bandwidth, and each *Mebeam* client generates much less traffic than the three other applications, it clarifies why *Mebeam* can support 16 simultaneous video users while *Nefsis* can only support 10 users, *Vsee* can support 8 users and *Qnext* can support 4 users.

5.3 Quality of Experience (QoE)

QoE can be measured through objective and subjective measurements. In this section, we assess via global measurements the QoE at the end user with respect to their video quality, audio-video synchronization, and level of interaction.

Video Quality In the objective measurements, we use bVQM (Batch Video Quality Metric) to analyze the VC's video quality off-line. bVQM takes the original video and the received video and produces quality scores that reflect the predicted fidelity of the impaired video with reference to its undistorted counterpart. The sampled video needs to be calibrated. The calibration consists of estimating and correcting the spatial and temporal shift of the processed video sequence with respect to the original video sequence. The final score is computed using a linear combination of parameters that describe perceptual changes in video quality by comparing features extracted from the processed video with those extracted from the original video. The bVQM score scales from 0 to approximately[10] 1. The smaller the score, the better the video quality.

We captured at every participant the stream from the imbedded multimedia player of each VC application with *Camtasia Studio 6*, and used[11] *VirtualDub* to cut and synchronize the frames of the compared videos.

Table 3 provides the bVQM scores for VC service per participant.

Table 3. The video quality of *Mebeam*, *Qnext*, *Vsee* and *Nefsis* at 4 clients.

VQM score	Client 1	Client 2	Client 3	Client 4	Average
Mebeam (Flash video, MPEG-4)	0.63	0.41	0.94	0.86	0.71
Qnext (MPEG-4, H.263, H.261)	1.05	0.94	0.63	0.83	0.86
Vsee	0.78	0.82	0.80	0.79	0.80
Nefsis (MPEG-4, H.263, H.263+)	0.34	0.61	0.61	0.87	0.61

Table 3 indicates that *Nefsis* features the best video quality among the 4 applications, although with an average bVQM score of 0.61 (its quality is only "fair", which will be explained later with the subjective measurements). The highest bVQM score (the worst video quality) appears at Client 1 (the super node) of *Qnext*. Generally speaking, all four VC applications do not provide good quality[12].

[10] According to [7], bVQM scores may occasionally exceed 1 for video scenes that are extremely distorted.

[11] *VirtualDub* is a video capture and video processing utility for Microsoft Windows.

[12] We also objectively measured the audio quality using metric PESQ-LQ (Perceptual Evaluation of Speech Quality-Listening Quality) [6] [8] and found that the PESQ-LQ

Because no standard has been set for what level of bVQM score corresponds to what level of perceived quality of a VC service, we have also conducted subjective measurements. We use the average Mean Opinion Score (MOS) [8], a measure for user perceived quality, defined on a five-point scale[13]: 5 = *excellent*, 4 = *good*, 3 = *fair*, 2 = *poor*, 1 = *bad*.

We gave 7 different quality videos generated by VC applications to 24 persons who gave a subjective MOS score independently. We also objectively computed their bVQM scores. Fig. 2 shows the correlation between the objective bVQM scores and the subjective MOS values.

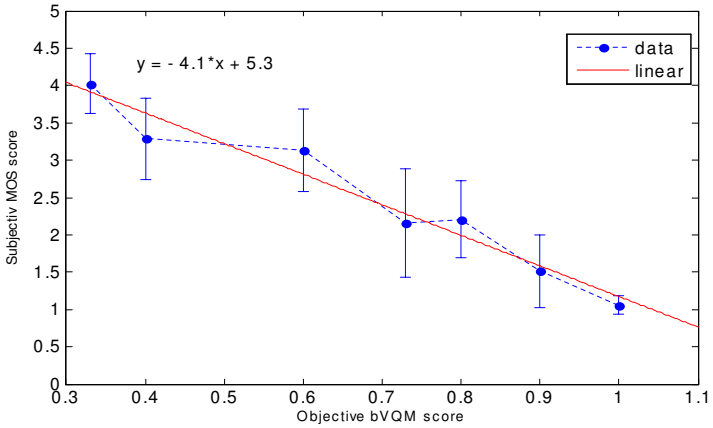


Fig. 2. Relation between bVQM and MOS for video conferencing service.

We mapped between the bVQM scores and the average MOS scores over 24 persons, and found that they have a linear correlation in the range 0.3 <bVQM score≤ 1. Hence, the VC's video quality is predictable when using the objective metric bVQM.

Compared with the video quality of a global P2PTV distribution service, which has an average MOS value of 4 [9], the video quality of a global VC service is poor (with an average bVQM score of 0.74 and MOS value of around 2.2), because the VC service requires end users to encode and upload their streams in real-time. Even the local uploaded video has a largely degraded quality although it is still the best among all participants.

---

average score (scale from 1.0 to 4.5, where 4.5 represents an excellent audio quality) is 2.24, 2.68, 3.08 and 3.15 for *Mebeam*, *Qnext*, *Vsee,* and *Nefsis,* respectively.

[13] The threshold for acceptable TV quality corresponds to the MOS value 3.5.

Audio-Video Synchronization The relative timing of sound and image portions of a streaming content may not be synchronized.

ITU [10] [11] has suggested that the viewer detection thresholds of audio-video lag are about +45 ms to −125 ms, and the acceptance thresholds are about +90 ms to −185 ms, for video broadcasting.

To analyze the A/V synchronization provided by each VC application, we used an "artificially generated" video test sample, in which the video and audio waveforms are temporally synchronized with markers. Similar to the experiments of testing the video quality, we captured at each end user the videos from all other participants. When the audio and video tracks were extracted and compared off-line, there was an average difference in time between the two tracks of about 650 ms for *Mebeam*, 470 ms for *Qnext*, 400 ms for *Vsee* and 350 ms for *Nefsis*. Such large audio-video lags are mainly caused by a large amount of frame losses, which lead to the low video quality mentioned already in Section 5.3.

Interactivity (communication delay) During a video conference it is annoying to have large communication delay[14]. Large communication delay implies lack of real-time interactivity in our global multi-party VC experiments. We measured the video delays among participants by injecting in the network another artificial video that mainly reproduced a timer with millisecond granularity.

In the video conference, this artificial "timer" video was uploaded via the virtual camera and transmitted among the participants via the different VC applications. At each participant, we used a standard universal Internet time as reference[15]. We displayed the "timer" videos of all participants in real time. After a 1-minute long stable video conference, we cut the captured content at each participant with *VirtualDub* to compare the "timers" between any 2 participants. For each application, we took samples at 2 different times to calculate an average delay.

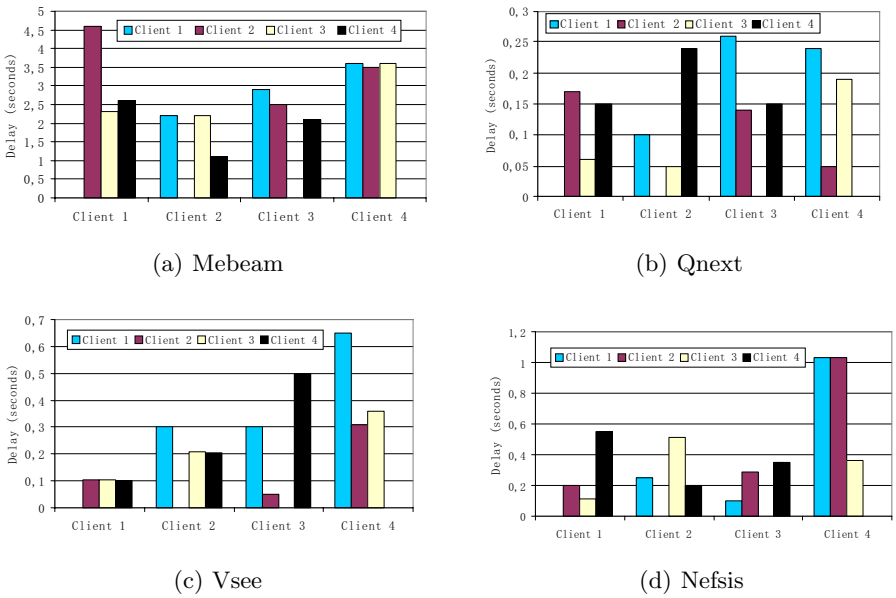
The video delays among participants are shown in Fig. 3. The x axis shows the 4 different clients. The y axis shows the video transmission delay from the participant on the x axis to the participant shown in the legend.

Fig. 3 shows that *Qnext* provides a video that is most synchronized among the clients. *Qnext*, *Vsee*, and *Nefsis* have a comparable level of average video delay, respectively 0.15 s, 0.27 s, and 0.41 s. However, *Mebeam* clients suffer a huge video delay (2.77 s on average), because the processing time at the server is too long.

We also measured the audio delays among participants by injecting in the video an artificial DTMF (Dual-tone multi-frequency) tone. We sent and recorded the audio at Client 1. Other participants kept their speaker and microphone on, but did not produce extra audio. Based on the recorded audio tracks, we compared the time the audio marker was sent from Client 1 and the time the same audio marker was heard again at Client 1 after the transmitted audio was

[14] In IP video conferencing scenarios, the maximum communication delay recommended by ITU is 400 ms [12].

[15] http://www.time.gov/timezone.cgi?Eastern/d/-5/java



(a) Mebeam (b) Qnext

(c) Vsee (d) Nefsis

Fig. 3. The video delay between different participants.

played, recorded, and retransmitted by a client. The time difference is approximately twice the one-way audio delay plus the processing delay at a client. Our results revealed 1 s, 1.4 s, 0.2 s and almost 0 s on average for *Mebeam*, *Qnext*, *Vsee* and *Nefsis* respectively. *Qnext* in this case provided the least synchronized audio among the users. When we measure the audio delay and the A/V synchronization, the delay is the end-to-end delay including the transmission delay and the delay introduced by the application. In our experiment, the video delay represents the delay of a same video scene that was captured at the application interfaces of the sender and the receiver, which does not include the time used for uploading the video to the sender via applications. Hence, considering the audio delay, video delay, and the A/V synchronization discussed in Section 5.3, we can conclude that the delay introduced by the application, when uploading, is large for *Qnext*.

6 Worst-case study

In another set of global experiments in June, 2009, our *Jperf* plots indicated that the end-to-end connections of clients 3 and 4 with the host were very unstable. We found that the two participants in China always passively disconnected from the conference or could not even log into *Mebeam*, *Nefsis* and *Qnext*. *Vsee* could still work, but the quality was awful, with bVQM scores close to 1.

In order to investigate the minimum bandwidth to support a video conference, we repeated many experiments adjusting the upload rate upper-bound (using[16] *Netlimiter*) at each participant for a particular VC application to test the user's upload bandwidth minimally required to launch a video conference.

For *Qnext*, the threshold is 50 Kbit/s. If an end user's available upload bandwidth is < 50 Kbit/s, (s)he cannot launch *Qnext*. For *Vsee*, the threshold is 50 Kbit/s; for *Nefsis* it is 28 Kbit/s; and for *Mebeam* it is 5 Kbit/s, which we believe are the minimally supported streaming bit rates set by the applications.

7 Summary and Conclusions

Through a series of local and global experiments with four representative video conferencing systems, we examined their behavior and mechanisms, and investigated their login process, the call establishment process, the packet size distribution, transfer protocols, traffic load, delivery topology, and different aspects of Quality of Experience.

Our main conclusions from the measurement results on the traffic characteristics of four different video conferencing systems are: (1) The QoE of multiparty video conferencing is very sensitive to bandwidth fluctuations, especially in the uplink. Hence, an adaptive bit rate/frame rate policy should be deployed; (2) When the number of participants increases, the traffic load at each participant does not always increase correspondingly (see Fig. 1), suggesting that re-encoding at the video takes place to limit the overall traffic in the system.

Our QoE measurement results are summarized as: (1) Existing Internet video conferencing applications in general cannot provide good quality to their end users (poor video and audio quality, large audio-video lag, and long communication delay in some cases); (2) Only a limited number of participants are supported and no high definition webcamera streaming is supported; (3) The existing systems are not reliable. When the network is unstable or the available upload bandwidth is very limited (thresholds have been found), none of the applications work properly.

It seems that the Server-to-Client architecture with many servers located all over the world is currently the best architecture for providing video conferencing via the Internet, because it introduces the least congestion at both servers and clients. Load balancing and load control algorithms help the overall performance of the system and the codec used is important for the quality that end users perceive. The bottleneck to support video conferencing with more participants and high definition streams is the overhead traffic generated by them. To support more simultaneous participants in a single conferencing session, the traffic load has to be controlled/limited by re-encoding the video streams.

We have chosen four representative video conferencing systems for our study, but the measurement methodologies mentioned in this paper can also be applied

[16] *NetLimiter* is an Internet traffic control and monitoring tool designed for setting download/upload transfer rate limits for applications.

to other video conferencing applications, which could be compared with our study in the future.

Acknowledgements

We would like to thank Rob Kooij for a fruitful discussion on measuring audio delay. This work has been supported by TRANS (http://www.trans-research.nl).

References

1. Salman Abdul Baset and Henning Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol", INFOCOM'06, Barcelona, Spain, April, 2006.
2. Luca De Cicco, Saverio Mascolo, and Vittorio Palmisano, "Skype Video Responsiveness to Bandwidth Variations", NOSSDAV'08, Braunschweig, Germany, May, 2008.
3. Richard Spiers and Neco Ventura, "An Evaluation of Architectures for IMS Based Video Conferencing", Technical Report of University of Cape Town, 2008.
4. Mark S. Silver, "Browser-based applications: popular but flawed?", Information Systems and E-Business Management, Vol. 4, No. 4, October, 2006.
5. Gregg Trueb, Suzanne Lammers, and Prasad Calyam, "High Definition Videoconferencing: Codec Performance, Security, and Collaboration Tools", REU Report, Ohio Supercomputer Center, USA, 2007.
6. Antony W. Rix, "A new PESQ-LQ scale to assist comparison between P.862 PESQ score and subjective MOS", ITU-T SG12 COM12-D86, May, 2003.
7. Margaret H Pinson and Stephen Wolf, "A New Standardized Method for Objectively Measuring Video Quality", IEEE Transactions on Broadcasting, Vol. 50, No. 3, pp. 312-322, Sep. 2004.
8. ITU-T Rec. P.800, "Methods for Subjective Determination of Transmission Quality", 1996.
9. Yue Lu, Benny Fallica, Fernando Kuipers, Rob Kooij, and Piet Van Mieghem,"Assessing the Quality of Experience of SopCast", International Journal of Internet Protocol Technology, Vol. 4, No. 1, pp. 11-23, 2009.
10. ITU BT.1359-1, "Relative timing of sound and vision for broadcasting", 1998.
11. Joseph L. Lias. "HDMI's Lip Sync and audio-video synchronization for broadcast and home video", Simplay Labs, LLC, August, 2008.
12. Ivano Bartoli, Giovanni Iacovoni, and Fabio Ubaldi, "A synchronization control scheme for Videoconferencing services", Journal of multimedia, Vol. 2, No. 4, August, 2007.