

User-interest Driven Video Adaptation for Collaborative Workspace Applications

Jeremiah Scholl, Stefan Elf, and Peter Parnes

Department of Computer Science & Electrical Engineering, Media Technology Division,
Luleå University of Technology,
971 87 Luleå, Sweden
{jeremiah.scholl, stefan.elf, peter.parnes}@cdt.luth.se
<http://media.sm.luth.se/>

Abstract. *This paper presents a bandwidth sharing scheme for group video conferencing. The key features of the scheme are the monitoring of implicit user-hints and message passing, which are used by each client in order to identify and report their interest in other group members. Information about user interest is then used by each video sender in order to adjust their frame-rate, resolution, and ultimately bandwidth consumption in an attempt to satisfy the current interests of the receivers. A general framework and initial prototype are presented together with experimental results.*

1 Introduction

Collaborative workspace applications based on IP-multicast offer a scalable solution for computer supported human-to-human communication. Even though Internet-wide multicast access is not a reality, these applications are making their way into work and educational environments due to the growing use of multicast in subnets, such as corporate and campus networks. While workspaces come in many flavors, the main function of these applications is to provide participating members with shared tools and a sense of presence. Video streams from the participating members are a key component of presence and multiple video streams can be resource intensive, making bandwidth sharing schemes for group video conferencing a key design component in applications of this type.

While many straight forward solutions can be applied, the human aspect and variety of contexts in which workspaces are used make designing a robust bandwidth-sharing scheme difficult. For example, the simplest solution is to allocate an equal portion of the session bandwidth to all users, but this will be inefficient when some participants in the group are of greater importance than others and could enhance the group's experience by using a larger share of available resources. An alternative approach that is suitable in some situations would be to allocate bandwidth based on floor control, but this is also inflexible as it can only be applied to settings where one participant should be the focus the group at any given time.

At Luleå University of Technology collaborative workspaces are used daily in a wide variety of situations. Example uses include allowing students to view classroom

lectures from home, enabling members of discussion groups to interact from distance, and providing members of projects and research divisions with increased presence of each other throughout their work day. The last mentioned case, referred to as the "e-corridor", can be viewed with particular interest in terms of bandwidth sharing due to the large variety of use-cases that it must serve. In the e-corridor members of the session can be active or passive and possible uses of the workspace range from giving or listening in on a formal presentation to passive monitoring of video. It may also include the occasional idle participant (i.e. out of the office). Thus, the number of "important" video streams in the session can vary widely over time.

This paper presents a bandwidth sharing scheme for group video conferencing aimed at a general use collaborative workspace environment, such as the e-corridor. The scheme operates by first identifying session participants that are of high importance to other group members and then allocating them a larger share of the session bandwidth. This is achieved primarily through the implicit detection of user-hints, such as the configuration of a user's desktop, and utilizes message passing so that receivers can reflect their interests back to the senders in question.

1.1 Related Work

The use of implicit user-hints in resource control has been applied to a wide range of multimedia applications, with each scheme being limited in scope to a specific domain. For example, Kulju et al. [1] investigated user hints in the context of video streaming, while Ott et al. [2] focused on its use within their own 3D landscape. In addition, recent work in collaborative workspaces has investigated how hints may be used in order to dynamically control the use of reliable multicast [3]. The work most similar to that presented in this paper is the SCUBA protocol [4], which also uses the detection of user interest in order to allocate bandwidth in video conferencing. SCUBA described the basic architectural components for schemes of this type, but little research has been done in this area since its introduction and new ideas as well as refinement on several points are still possible.

2 Design of a User-Interest Driven Video Adaptation Scheme

The bandwidth sharing scheme described in this section follows the same architecture as SCUBA, but also differs from it in several ways. The first is that a novel approach for bandwidth sharing is used that seeks to first fulfill the minimal needs of all senders before dividing the remaining bandwidth among important group members. In addition, information about user interest is used to help each sender select the correct parameters in the tradeoff between image resolution and frame rate, which is something that SCUBA does not take into consideration. Another key difference is that optimizations for message passing are presented in the context of empirical observations made about how humans interact with collaborative workspaces, whereas SCUBA presented an alternative method based on statistical sampling. Finally, greater flexibility is shown in the number and types of user-hints explored.



Fig. 1. Video windows included in the Marratech Work Environment

In order to determine the video streams that are of interest to a particular receiver one must answer the question, *”Who is this user currently viewing, and in what context?”*

In regards to viewing context, the range of possible answers is defined by the video windows provided inside the application user interface. The Marratech Work Environment [5], which we have used for prototyping, is shown in Fig. 1 and includes video panels in several different windows. These windows are designed to compliment each other and allow each participant to view other members in a variety of ways. Similar to the well known research application vic [6], Marratech provides users with a *”Participants”* window, which gives a thumbnail overview of the video streams currently received from the group, and a *”Focus”* window that displays the video obtained from a single group member at higher resolution. In addition to these windows, Marratech also contains a small video panel in each private chat display, which allows two participants to easily obtain response clues such as posturing and facial gestures (smiling etc.) while chatting in private.

Table 1. Video windows in Marratech client.

Window	Purpose of Panel	Pixel Resolution	Min. Frame Rate
Focus	high level of presence	702 x 576, 352 x 288, 176 x 144	5 fps
Private Chat	response clues	176 x 144, 88 x 72	1 fps
Participants	overview of activity	88 x 72	.2 fps

Table 1 lists each of these windows specific role in presence delivery along with their individual resource requirements including resolution and minimum "acceptable" frame rate, which was obtained from a variety of sources. In regards to the Focus window, the minimum acceptable rate is derived from the various work summarized by Chen [7]. This includes work by Tang et al. [8], who noted that users consider 5 fps to be tolerable, and Watson et al. [9], who found that users do not perceive audio and video to be synchronized at frame rates lower than 5 fps. Other studies have also shown little difference in communication behavior or task outcome between 5 fps and 25 fps [10], [11], [12]. Together this work suggests that 5 fps will provide users with an adequate experience in a variety of situations requiring a high amount of attention. The values provided for the Focus window and private chat windows were obtained through a survey conducted of expert Marratech users, and reflect the values most typically reported as "tolerable" for the each of the respective windows.

2.1 Identifying Important Video-Streams

The primary method for detecting user interest is to monitor user interface parameters that will reveal the video senders currently loaded in each of the video panels described above. In our case, this leads to a host giving one of four possible classifications to each sender, one for each of the separate video window configurations and one classification for members that are currently not viewed in any available panel. For some applications it may also be desirable to create classifications that describe senders contained in multiple panels simultaneously, but with the Marratech environment this is not necessary because the frame rate and resolution required for panels delivering a high level of presence will also be sufficient for each lower level. Thus, a video stream that is delivered for the Focus window will also be sufficient for the Participants window and so on.

Cross-media clues can also be used to detect an important video stream [4] with the most useful example being the monitoring of audio. The current audio sender is usually a leading presenter or an otherwise important participant in group discussions so the Marratech application gives users the option of selecting "video follows audio", which will automatically move the current speaker into the Focus window. Monitoring the content of the Focus window will still be sufficient to detect an important stream in this case, but the audio clue can be useful to reduce the latency it takes for a sender to realize its importance and to further prioritize audio senders over other "focused" participants as described in the next subsection.

The whiteboard and chat can also provide useful clues, but of a somewhat different nature than audio and video. While drawing with the whiteboard pen or sending a chat message may be a sign that a user has become interesting to other users, this will likely only be for a short period of time while they "check out" the user's activity. Therefore, when a sender has a low frame rate (under 1 fps) an event from either of these media can be used in order to have him send an extra frame or two.

Downgrading a Sender: At times user interface monitoring and cross-media clues can be misleading and may cause a client to identify senders as important when in fact their video feeds are expendable. E-corridor participants can for example typically leave their

office for an extended period of time, which may result in a client that continues to act on the behalf of its user even though no one is in the room to view the video streams received. One strategy that can be adopted in order to minimize the impact from this type of misidentification is to obtain hints regarding events external to the application before making decisions on behalf of the client. Hints of this type work to downgrade a sender that would otherwise be identified as important, and can further refine the process of detecting user importance. Several example hints in this category are listed below.

Detecting Idle Receivers It is pointless for a receiver to continue requesting video from senders when no one is actively using the computer. One primary method for detecting an idle receiver is to monitor the user's screen saver. This can be complemented by other techniques, such as the monitoring of peripheral input devices like the keyboard and mouse, and/or the detection of a lack of movement in front of the user's camera.

Window placement When windows from other applications cover up a video panel, it is a solid indication that the user is not interested in the incoming video stream [1]. This should also be true if the video window in question is minimized.

Limited Resources Even if a user can benefit from receiving additional data it does not guarantee that he has enough resources to do so. This can be especially true when using a mobile client as they are often more limited by CPU and memory resources than available bandwidth.

2.2 Video Adjustment Algorithm

The video adjustment algorithm we have designed works first to provide each sender with the minimum acceptable frame rate and proper image resolution for its most interested receiver, with unused bandwidth beyond that point distributed evenly among the highest priority senders in the group. The rationale for using this "minimum requirements first" strategy is that it allows important senders to deliver the richest experience possible while keeping them from punishing less important senders. The main drawback of this method is that it may not be appropriate for use with sessions that have very limited bandwidth, for example those which intend to support modem users, because the aggregate requirements of even the least demanding senders may be hard to meet. However, sessions of this type are today generally viewed as a special case and most likely need a scheme that is optimized specifically for use with low bandwidth sessions [7], rather than a scheme that is designed for general use like the scheme that is described in this paper.

Obviously it is not realistic to assume that the minimum acceptable requirements will be the same in every situation. In practice the administrator of the session should have the option of setting these values. However, in order to make the creation of sessions more user friendly it is important to have a workable set of default values that can be used when the administrator does not exercise this option. With this in mind we have done an analysis of expected bandwidth usage when sending at several of the appropriate frame rates and image resolutions discussed in Table 1.

Table 2. Estimated bandwidth usage for each sender

Window	Frame Rate	Resolution	Bandwidth Usage
Focus	5 fps	352 x 288	55 kb/s - 160 kb/s
Private Chat	1 fps	88 x 72	8 kb/s - 20 kb/s
Participants	.2 fps	88 x 72	less than 1 kb/s - 4 kb/s

Table 2 includes this information and can be used as a reference when trying to determine how well a minimum requirements approach will scale in the real world. The bandwidth measurements included were taken from a Marratech e-meeting client while sending video data at various frame rates and resolutions included in Table 1. The fourth column in Table 2 shows bandwidth measurements taken during "typical" use, with the low value representative of users that are fairly still in front of their computer, and the high value taken during moments of high activity, such as the user moving about or interacting with another person in the office. It should be mentioned that although the private chat and Focus windows have variable resolutions our measurements were taken with the default settings applied.

The numbers in Table 2 should only be treated as estimates, as variations in bandwidth consumption can be expected due to real-world factors, such as the camera type in use and the amount of motion between frames. They do however show that for a typical session (less than 50 users) it is not difficult to meet the minimum requirements for the Participants window due to the low bandwidth required by each sender. In practice this is also true for private-chat users because the concurrent number of chats is usually equal to a small fraction of the number of session participants. However, the requirements of each "more important" sender, defined as those currently sending audio or being viewed in the Focus window, may be difficult to meet if the attention of the group is too "spread out" or if the session has low to medium available bandwidth (256 Kb/s - 500 Kb/s).

Each sender operates within the scheme by classifying itself on a scale from 0 to 4 based on how it is viewed by other group members and whether or not it is currently sending audio. These classifications are: 4 - audio sender; 3 - Focus-window sender; 2 private-chat sender; 1 Participants-window sender; and 0 - no interested receivers. A host uses information about its class in order to determine its frame rate and resolution as given in Table 1, and measures the incoming bandwidth consumption of other members in order to determine the amount of bandwidth available to it. This information is then used by the sender in order to adapt its video using the priority scheme described below.

Step 1: Bandwidth is divided evenly between all the senders until each sender can send at the minimum frame rate and resolution for the Participants window.

Step 2: If there is still session bandwidth available after step 1, it is allocated between the senders of class 2 or higher until they are sending at the minimal frame rate and resolution for the private chat window.

Step 3: If there is still available bandwidth after step 2, it is divided between senders of

class 3 or higher until they can send at the necessary frame rate and resolution for the Focus window. This is done first for class 4 senders, and then for class 3 senders.

Step 4: All remaining bandwidth is divided evenly between each class 3 and 4 sender.

2.3 Receiver Feedback

In order for a sender to be aware of how it is viewed by other group members a mechanism needs to be in place that allows each receiving host to communicate their interests via messages. The simplest way to do this is to have each receiver automatically send a message each time an event occurs that causes it to reclassify a sender. This approach may of course end up in unnecessary messages being passed but it is not clear if this will consume enough bandwidth to significantly reduce the performance of the application.

Several techniques can be applied in order to reduce the number of unnecessary messages with the most obvious example occurring when someone starts to send audio, which will cause them to be moved into the Focus window by several participants simultaneously. In this situation a more efficient approach than having each receiver send a message is to instead have each receiver inform the group when they change the "video follows audio" option, which will enable the accurate use of the audio clue mentioned in sect. 2.1.

Table 3. User interactions logged during empirical study and influence on sender class

Event	Bandwidth class	Up- or Down-grading
Un-muting audio	4	Up
Muting audio	4	Down
Viewing or maximizing main video	3	Up
Minimizing or closing main video	3	Down
Opening private media	2	Up
Closing private media	2	Down
Un-muting participant video	1	Up
Muting participant video	1	Down

An unnecessary message may also be created when a receiver views a sender in a new context while already receiving enough video. For example, if a sender has a frame rate of 5 fps due to the actions of other receivers, it is pointless to send it a message when opening up a private chat window, as this requires a refresh rate of only 1 fps. Messages of this type can be reduced by having each receiver monitor the frame rate and resolution of incoming streams and pass messages only when they are deemed to be inadequate. The drawback of this technique is that it will make it difficult for senders to know exactly whom every receiver is watching, and will thus require an additional mechanism so that each sender can find out when they should reduce their bandwidth after receivers have lost interest in them. A simple way to handle this is to include

information about how each sender is viewed in RTCP receiver reports, which will solve the problem but will also introduce latency in the bandwidth reduction process. SCUBA takes a different approach towards feedback and uses statistical sampling rather than obtaining messages from the entire group. The advantage of this method is that it improves overall scalability because the number of messages grows logarithmically rather than linearly as the session size increases.

Because messages should only be created based on specific user interactions it is not clear if any of the above message reduction strategies are necessary, or if these interactions will typically be infrequent enough to make the number of messages passed in the session negligible. In order to gain further understanding of the potential amount of bandwidth that messages may consume, we conducted an empirical study of a research group consisting of nine daily Marratech users. This was done by creating a prototype version of Marratech, which generates messages based on specific user interactions as summarized in Table 3, and distributing it among these users. The messages were logged over a three day period under normal working conditions, which included a formal research discussion on the last day, as well as periods of more "common" use.

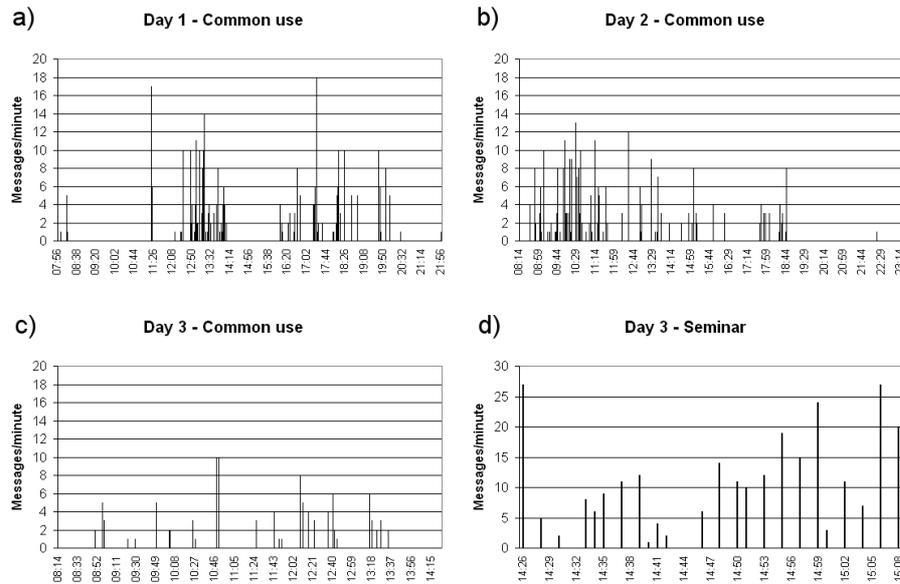


Fig. 2. Graphs showing the total number of interactions within a 9 user research group over a three day period. (Note the scale difference in (d).)

Figure 2 shows four graphs of activity during the logging period, in which a total of 1046 interactions were detected, corresponding to 119244 bytes worth of data. The average message length was 114 bytes, which included a sender id, timestamp and an indication of the interaction in question. It should be noted that these messages were

not optimized in any way, so in practice it should be possible to reduce this size. Graphs **a**, **b** and **c** show activity during "common use" periods and are highlighted by a fairly low amount of activity, with some short bursts occurring that correspond to increased interaction between the users.

As expected the most intensive period of message creation by far occurred during the research discussion on the last day, which is shown in graph **d**. This included the hour of highest activity during the three days, in which 380 messages were sent. The peak minute of usage during the research discussion resulted in a total of 27 interactions, which corresponds to an average bandwidth consumption of less than .05 KB/s and a total bandwidth consumption of 3068 bytes. This shows that even if all the messages during the most active minute of the discussion were created simultaneously that the amount of bandwidth consumed would be negligible.

Thus, a conclusion can be drawn from this study that during normal use the total number of messages expected should consume a tiny portion of the session bandwidth, even if no optimizations are in place.

3 Summary and Conclusions

We have introduced a framework for bandwidth sharing in video conferencing that uses the implicit detection of user interest as a metric for resource allocation. Schemes of this type contain three architectural components, which are the detection of user interest, message passing, and bandwidth adjustment algorithms. In the area of user-interest detection we have described several methods for identifying users' interests and have introduced new ways to reduce the number of false positives in this process. In addition, we have expanded the area of bandwidth adjustment in order to help senders correctly identify their optimal frame rate and image resolution in each situation and have done so by adopting a "minimum requirements first" strategy. This strategy attempts to provide each sender with the minimum frame rate and image resolution for its most interested receiver before assigning the remaining session bandwidth to the senders deemed to be most important.

We have also discussed several different mechanisms designed to reduce the number messages created, and have conducted an empirical study in order to determine how necessary they are during real use. This study was conducted by deploying a prototype we created among a research group at our university that allowed us to monitor the messages they generated. We concluded from this study that, given the interactions from Table 3, messages will occur infrequently enough during normal use that such message reduction mechanisms are of little use in practice, even though they may be academically interesting.

3.1 Future Work

Our first priority in the future is to make a complete, user friendly prototype that can be distributed among the Marratech test users. This will require us to look into several issues including user-interface options so that users can "opt out" of the dynamic bandwidth process. In the real world this will be necessary because there are some situations

when it is most beneficial to allow certain users in the session to set their bandwidth consumption manually. We also plan to study the performance of our scheme in this type of mixed environment.

In addition, robust user studies are needed in order to find further ways of refining the scheme. In particular, it is not clear at this time if it is best to divide all the extra bandwidth between only the important senders in the group as stated in sec. 2.2, or if there is a more optimal strategy. In some situations for example it may be better for a portion of the extra bandwidth to be used in order to increase the frame rate of clients in the Participants window.

4 Acknowledgments

This work is supported the Centre for Distance-spanning Technology, the Swedish Research Institute for Information Technology, the VITAL project and the Mäkitalo Research Centre. Stefan Elf is also with Ericsson AB, SE-931 87 Skellefteå, Sweden. Views expressed in this paper are his own and not necessarily shared by his employer.

References

1. Kulju, W., Lutfiyya, H.: Design and implementation of an application layer protocol for reducing udp traffic based on user hints and policies. In: 5th IFIP/IEEE International Conference on Management of Multimedia Networks and Services, MMNS 2002. (2002)
2. Ott, M., Michelitsch, G., Reiningger, D., Welling, G.: An architecture for adaptive qos and its application to multimedia systems design. Special Issue of Computer Communications on Guiding Quality of Service into Distributed Systems (1997)
3. Scholl, J., Elf, S., Parnes, P.: Efficient workspaces through semantic reliability. In: 10th International Conference on Telecommunications, ICT 2003. (2003)
4. Amir, E., McCanne, S., Katz, R.H.: Receiver-driven bandwidth adaptation for light-weight sessions. In: ACM Multimedia. (1997) 415–426
5. Marratech: - The e-meeting company. URL¹ (2003) Visited March 30th, 2003.
6. McCanne, S., Jacobson, V.: vic : A flexible framework for packet video. In: ACM Multimedia. (1995) 511–522
7. Chen, M.: Achieving effective floor control with a low-bandwidth gesture-sensitive video-conferencing system. In: ACM Multimedia. (2002)
8. Tang, J.C., Isaacs, E.A.: Why do users like video? studies of multimedia-supported collaboration. *Computer-Supported Cooperative Work: An International Journal* (1993) 163–196
9. Watson, A., Sasse, M.A.: Evaluating audio and video quality in low-cost multimedia conferencing systems. *Interacting with Computers* **8** (1996) 255–275
10. Jackson, M., Anderson, A.H., McEwan, R., Mullin, J.: Impact of video frame rate on communicative behaviour in two and four party groups. In: *ACM Computer Supported Cooperative Work*. (2000) 11 – 20
11. Masoodian, M., Apperly, M.: Video support for shared work-space interaction: An empirical study. *Interacting with Computers* **7** (1995) 237–253
12. Ghinea, G., Thomas, J.: Qos impact on user perception and understanding of multimedia video clips. In: *ACM Multimedia*. (1998)

¹ <<http://www.marratech.com>>