# Selecting the QoS Parameters for Multicast Applications Based on User Profile and Device Capability

Khalil El-Khatib, Gregor v. Bochmann, and Yu Zhong

School of Information Technology & Engineering, University of Ottawa
161 Louis Pasteur St., Ottawa, Ont., K1N 6N5, Canada
{elkhatib, bochmann, yzhong}@site.uottawa.ca

**Abstract**. Most adaptive multimedia multicast applications require the source to select the number of streams to transmit as well as the QoS parameters for each stream. If the receivers have different bandwidth limits for their devices and have various preferences for the quality of the data, selecting the QoS parameters that generate the best average satisfaction for all receivers is a challenging problem. In this paper, we developed a selection algorithm that is based on the user profiles and the device capabilities. Receivers are required to send their profiles and the bandwidth limitation on their devices to the source once before the session starts. To avoid the implosion problem and have a constant running time for the selection algorithm, we partition the receivers according to the bandwidth limit of their devices into classes and use a virtual representative for each class of receivers.

## 1 Introduction

Multimedia broadcast applications like tele-teaching, teleconferencing, Internet TV, or remote presentation, are becoming valuable services for users of the Internet. These applications became possible due to advances in the capabilities of desktop machines and transport networks. At the level of transport, the most important advance that made these applications possible was the development of broadcast protocols, such as the Multicast Backbone (Mbone). The Mbone is an experimental virtual network imposed on the top of the Internet and has been used as a multicast test-bed. The Mbone consists of islands of multicast-capable networks, connected to each other by virtual links called "tunnels", and it shares the same transport infrastructure as the Internet.

Adaptive multicast applications require that the server solicit information from all its receivers in order to adjust the sending parameters as to provide the highest possible level of service to the receivers. When receivers are connected through variable bit rate connections (over the Internet or a Variable Bit Rate (VBR) ATM feed), receivers are usually required to periodically send feedback messages to the source in order to accommodate the changes in the available bandwidth along the path from the source to the receiver. Most of the works in the literature have focused on sending the data loss rate at the receiver as an indication of the congestion along the data path.

D. Shepherd et al. (Eds.): IDMS 2001, LNCS 2158, pp. 221-232, 2001.

Based on the feedback reports, the source might adjust the sending rate [7,10,11] or keep the same sending rate and send more error-resilient data [1]. As the number of feedback messages scales with the number of receivers, with large numbers of receivers, the source might not be able to handle all these feedback reports, especially when these reports are sent periodically. This problem is known in the literature as implosion. Several approaches have been proposed to solve the implosion problem including back-off timers [2, 3], probabilistic polling [4], and network aggregation [5].

In our work, we simplify the problem of implosion by assuming that the receivers will only send one report before the session starts. Information in the report includes the preferences of the receiver for the QoS parameters such as frame rate and resolution and the bandwidth limit of the available local network access link. This information is used by the server to compute the QoS parameters for each stream to send. Our approach differs from other research work in that the selection of the QoS parameters is based on the user preferences (frame rate and resolution) and the limitation on the bandwidth of his device rather than the loss rate. Receivers are classified according to their bandwidth limit. Based on the capability of the source, our approach selects the number of streams as well as the QoS parameters of each stream. If the source has enough throughput limit to send a separate stream for each class of users, the source runs the selection algorithm for each class separately. If not, the source has to decide then on the number of streams to send as well as the QoS parameters for each stream. In both cases, the selected QoS parameters ensure the highest level of satisfaction of the receivers within their preferences, their device capability and the throughput limitation of the source. To avoid the scalability problem with the number of receivers, we introduced a way to select a virtual representative for each class, representing all receivers in the class. Receivers that haves the same bandwidth limit are more likely to have similar QoS preferences and adequately represented by one virtual receiver.

The rest of the paper is organized as follows: Section 2 gives a literature review of adaptive multicast applications. Section 3 introduces the user profiles and device capabilities used in the selection algorithm presented in Section 4. Based on the throughput limit of the source, Section 4 details two algorithms to select to QoS parameters for each stream to send. The use of group representatives is outlined in Section 5, and finally, a conclusion in Section 6.

## 2 Literature Review

Adaptive multicast applications are classified as either sender-driven or receiver-driven applications. In the sender-driven approach, the congestion feedback messages from the receivers and/or network are used to adapt the transmission rate at the source. In the receiver-driven approach, the source transmits the same data in different variants, and the receivers select the variant to receive depending on the congestion they encounter.

In [4], a single video stream is transmitted to all receivers, and congestion feedback is used to control the rate of the video stream. To prevent feedback implosion, a form of probabilistic feedback is used. In [6] the authors proposed an approach for the design of an available bit rate congestion control algorithm that maximizes inter-receiver fairness for multicast Available Bit Rate (ABR) sessions. Each receiver is assigned a weight value, and has an “isolated bandwidth” defined as the rate that would be achieved by the receiver when it is the only receiver in the multicast group. Every single receiver defines his maximum acceptable loss tolerance *l* and selects its own “receiver fairness function” that maps from the actual operating bandwidth value to a fairness value. The sender receives a feedback from each receiver including the isolated bandwidth and the receiver fairness function. The sender will then try to determine the sending rate that maximizes the weighted fairness among receivers. To achieve scalability, intermediate nodes are used to aggregate feedback messages from receivers according to their isolated rates. This approach was modified for use in the Internet [7]. The authors used the loss rate instead of the isolated rate to find the best sending parameters for the stream. The authors also suggested the use of two streams: a base stream with a constant data rate that can accommodate the receivers with the lowest connection rate, and another variable bit rate stream whose data rate can be modified based on feedback messages from receivers. Receivers with requests to lower the bit rate of the V-stream can always change to the base stream.

Layered encoding and group multicasting are combined in the receiver-driven approach. In the receiver-driven layered multicast (LRM) [8] approach, the video stream is decoded as a “basic” video stream, and a set of enhancement layers. Each layer is sent to a different multicast address. Receivers should receive the basic video stream and the enhancement layers that best suit their requirements. A receiver may use join-experiments to add more layers when there is extra capacity and release layers when the receiver experiences congestion. Thin Streams [9] reduces the congestion resulting from the join-experiments by dividing each layer further into “thin” layers.

The Destination Set Grouping (DSG)[10] is a hybrid between sender-driven and receiver-driven approaches. In this scheme, the source maintains a small number of video streams, broadcasting different variants of the same information. Receivers can tune to the stream with the quality they prefer. They can also send feedback messages to the source to adjust the quality of the stream to which they are tuning. The DSG protocol is composed of two components: an *intra-stream* protocol and an *inter-stream* protocol. Using the *intra-stream* protocol, receivers can determine their status as: LOADED, UNLOADED, and CONGESTED depending on the packet loss rate. Receivers are polled in a probabilistic manner in order to estimate the number of CONGESTED and UNLOADED receivers. Depending on the fraction of CONGESTED receivers, the source adjusts the sending rate in order to keep most of the receiver in the LOADED state.

The Source Adaptive Multi-Layered Multicast (SAMM)[11] is another hybrid algorithm that uses congestion feedback from the receivers to adjust the number of the generated layers as well as the encoding parameters of each layer. Two variations of the algorithm were proposed: a network based SAMM algorithm and an end-to-end SAMM algorithm. In the network-based algorithm, the source periodically generates a control packet called “forward feedback packet” and sends it to the multicast group. Each intermediate node updates the packet with the amount of bandwidth available for the transmission of the multicast flow. When the packet arrives at the receivers, it contains the bandwidth available on the path from the source to the receivers. Each receiver stores this value in a feedback message and sends it back to the source. Intermediate mergers combine feedback messages from downstream nodes and forward only one feedback message toward the source. If the number of requested rate values is higher than the maximum number of video layers allowed, then one or more rates must be discarded. The algorithm drops the layer with the smallest number of receivers and adds the number of receiver of that layer to the number of the preceding lower layer. The end-to-end algorithm is similar to the network-based algorithm, except that the receivers cannot determine their available bandwidth, and they only use an estimate based on the received video rate. Because the actual available bandwidth could be higher than the video arrival rate, the receiver might occasionally report a rate that is higher than the observed rate.

Active networks can also be used with adaptive multicast application [12]. In this approach, trans-coders are installed at intermediate nodes in the multicast distribution tree. The source sends only one high rate variant of the data, and intermediate nodes do the trans-coding depending on the requirements of down-stream receivers and network congestion. While this approach distributes the overhead of the source, saves bandwidth by sending only one variant of the data, and performs trans-coding only when necessary, the control and management of intermediate trans-coders is a complex problem.

## 3 User Profiles and Device Limitations

In earlier work [13], we used the DMIF session management protocol to develop a distributed QoS management framework for multicasting multimedia applications. The protocol aimed at distributing part of the QoS management process between source and receivers; each receiver process can make certain QoS decisions based on its local context. The QoS manager in the source node determines the list of potential stream variants for each logical multimedia stream, and informs all the receivers about these variants. Based on the user profile, the QoS agent at the receiver node selects the stream that gives the highest level of appreciation to the receiver. The QoS agent can request a certain stream from the QoS manager if the stream is not currently transmitted. We used the control-plane of DMIF for the session management and illustrated its usage for the management of a tele-teaching application including different QoS alternatives for the participating users.

In other work [14], we presented an architecture for personal telecommunication services, in which each user was required to have a profile that contains all his personal information. This information covers QoS preferences for multimedia communication in addition to the user policies for handling incoming and outgoing calls. Based on this information in the profile of all communicating parties in the session, and based on the limitations of their candidate devices, we presented a middleware architecture that can select the devices and QoS parameters for the session that best suit all the parties.

For the adaptive multicast applications discussed in this paper, we will restrict the user profile to the user's QoS preferences. As proposed in [15], we assume that each user specifies the *minimum acceptable* and the *ideal* value for each QoS parameter (such as frame rate and resolution). A satisfaction function that maps the actual QoS value of the user satisfaction onto a range between 0 and 1 is shown in figure 1. QoS parameters that are higher than the *ideal* values in the user's profile generate the same level of the satisfaction as the *ideal* values. The total satisfaction of the user is computed as the weighted average of satisfactions of individual QoS parameters (see [15] for more details).

Another important parameter is the bandwidth limit of the device that is used to receive the multicast stream. This value determines the class to which the receiver belongs (see Section 5 for more details).

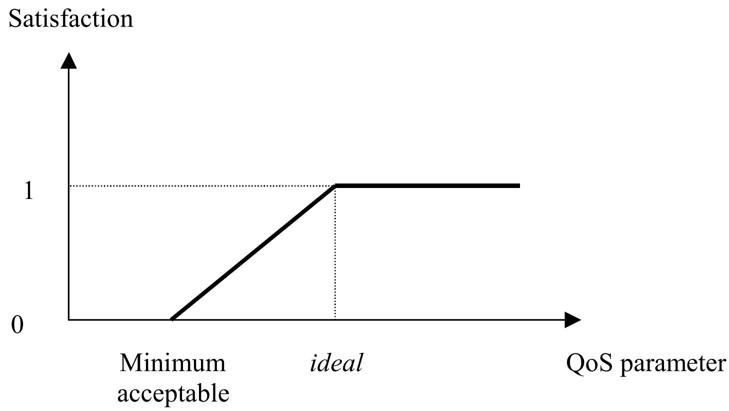


**Figure 1.** User satisfaction for a QoS parameter

The combination of user QoS preferences and the bandwidth limit of his device determine the possible parameters that the receivers can accept. Any accepted combination of QoS parameters must have each individual parameter higher than the *minimum accepted* parameter, and the combination must have bandwidth requirement less than the bandwidth limit of the device. A combination of QoS that requires a bandwidth limit higher than the bandwidth limit of the device will cause some packet loss and hence a zero (0) satisfaction value for the receiver.

## 4 Selecting QoS Parameters for Large Groups of Users

Traditional QoS parameter selection has been addressed in the context of small groups, mostly in two-party sessions or in a small group of users collaborating in a teleconference session. With such small group, it is feasible for group members to negotiate the QoS parameters that represent best the participants' preferences. However, the negotiation algorithms for small groups do not easily scale to large groups. The task of finding a common denominator for all participants in a large group can pose interesting and challenging technical problems.

An approach to find a common ground for all participants in a session is to send the preferences of all participants to one node, which tries to find the QoS parameter values that generate the maximum satisfaction among all participants. In our approach, we assume that this node is the multimedia server or any node that controls the sending parameters of the source. All candidate receivers are required to send their profiles to this node. Intermediate nodes may be used to aggregate the user profiles as explained in Section 5. As we mentioned earlier, each profile will include, in addition to the QoS preferences of the user, the available bandwidth to the receiver. We assume that this available bandwidth is known and remains constant throughout the session.

In our earlier work on QoS management for inter-personal communication [18], we have adopted the framework presented in [15] for selecting the QoS parameters that provide the highest satisfaction for the user. The user satisfaction for a single application QoS parameter ($S_{Xi}$) is computed based on the *minimum accepted* ($M_{Xi}$) and *ideal* ($I_{Xi}$) values defined by the user profile. The user satisfaction ($S_{user}$) for a combination of different QoS parameters is computed as a function of the satisfaction for the individual parameters. We have modified the framework to assign weights to individual parameters. In addition, we have extended the framework to consider the overall satisfaction obtained for a collection of several users participating in the application. The total satisfaction of all users ($S_{total}$) is also computed as a function of the satisfactions of all the users, including possibly weight factors. The functions used to calculate $S_{user}$ and $S_{total}$ share the property that their value becomes zero when one of the individual constituent satisfactions becomes zero. This property ensures that all participants in a session will obtain a quality that is better than the *minimum acceptable* value. We note that this property is not ensured in the case that a simple weighted average is used as the function to combine the individual satisfaction values. The definition of $S_{Xi}$, $S_{user}$ and $S_{total}$ is the following:

$$S_{Xi}(x) = \begin{cases} 0 & if\ x < M_{Xi} \\ 1 & if\ x > I_{Xi} \\ \dfrac{x - M_{Xi}}{I_{Xi} - M_{Xi}} & otherwise \end{cases} \tag{1}$$

$$S_{user} = f_{comb}(s_{X1}, s_{X2}, s_{X3} \ldots, s_{Xn}, w_1, w_2, w_3 \ldots, w_n) = \frac{n\overline{w}}{\sum_{i=1}^{n} \frac{w_i}{s_{Xi}}} \tag{2}$$

$$S_{total} = f_{comb}(s_{user1}, s_{user2}, s_{user3} \ldots, s_{userM}, w_{user1}, w_{user2}, w_{user3} \ldots, w_{userM}) = \frac{M\overline{w}}{\sum_{i=1}^{m} \frac{w_{user\_i}}{s_{user\_i}}} \tag{3}$$

In the case of multicasting, we found that this framework for computing the overall satisfaction is not appropriate, since with a large population of receivers, at least one of the receivers would have a zero value for his satisfaction for any combination of QoS parameters, and therefore the overall satisfaction would always be zero. Instead, we decided to use the simple weighted average to determine the overall satisfaction of all receivers. Candidate receivers that have a zero satisfaction for the select QoS parameters simply will not join the session. This function for $S_{total}$ can be written as follows:

$$S_{total} = f_{comb}(s_{user1}, s_{user2}, s_{user3} \ldots, s_{userM}, w_{user1}, w_{user2}, w_{user3} \ldots, w_{userM}) = \frac{\sum_{users} w_{user\_i} s_{user\_i}}{\sum_{users} w_{user\_i}} \quad (4)$$

### 4.1 Selecting QoS Parameters with Unlimited Throughput in the Source

Before the session starts, all receivers are required to send their reports to the source. The source then classifies the receivers into separate classes according to their bandwidth limits. If the source has throughput limit higher than the sum of all bandwidth limits of the class, it tries to find the QoS parameters for each class separately. The next section deals with the case when the server has a throughput lower than the throughput required by all classes.

For each class, the source tries to find the combination of the QoS parameters that generates the highest average satisfaction of all receivers in the class. This combination must also require lower bandwidth than the bandwidth limit of the class. The source selects the QoS parameters for every class of receivers, based on the preferences of the receivers in the class and their bandwidth limit. The source sends after a multicast report to all receivers, informing them of the number of streams, the QoS parameters for each stream, and the multicast address for each stream. Each receiver determines the stream that best suits his preferences and tunes to the address of that stream, as described in our earlier work [13].

### 4.2 Selecting QoS Parameters with Limited Throughput in the Source

It is very possible that the source receives requests to deliver a number of streams that exceed its bandwidth throughput. In this case, the source has to decide on the number of streams to send, as well as the combination of the QoS parameters for each stream.

To do the selection, the source does not separate receivers into separate classes as in the case with unlimited throughput, but it keeps all receivers in one group. When using class representative as discussed in Section 5, receivers of the same class are represented as one virtual representative, and all class representatives are treated together in one group. The source then tries to select the combination of possible number of streams and the QoS parameters of each stream. The combination that generates the highest average satisfaction is the one adopted by the source.

We have done some experiments using the MATLAB Optimization Toolbox 2.1 to see how the average satisfaction of the receivers and the number of streams change as a function of the throughput limit of the server. We selected four bandwidth limits (128Kbps, 256Kbps, 512 Kbps and *unlimited*) on receivers (four classes), and we run the experiment with 1000 receivers. The user preferences (i.e. the *minimum accepted* ($M_{Xi}$) and *ideal* ($I_{Xi}$) values) were selected randomly with uniform distributions for $M_{Xi}$ and $I_{Xi}$ within certain ranges. These ranges were chosen in such a manner that the ideal quality for most users would correspond to a bandwidth beyond the bandwidth limit of that particular user (users are usually optimistic about the quality they might receive).

The results of these simulations are shown in figure 2. The graph shows clearly that as the bandwidth limit of the source increases, the number of streams and the average satisfaction increases until it reaches a maximum point. While the average satisfaction did not get to one (1), all users were receiving up to the maximum bandwidth of their devices. Figure 3 shows how the number of streams increased from one to four, leading to higher average satisfaction for all receivers.

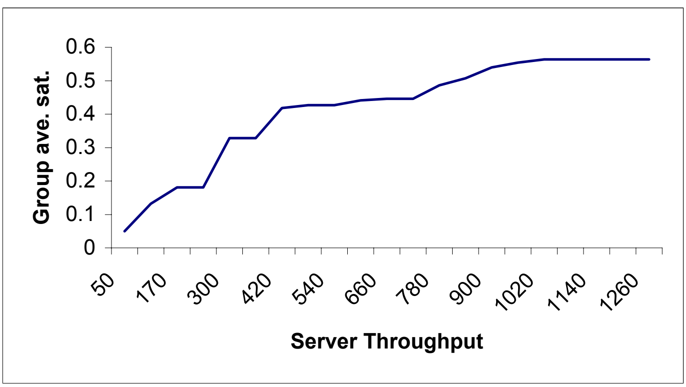


**Figure 2.** Server bandwidth limit vs. average satisfaction

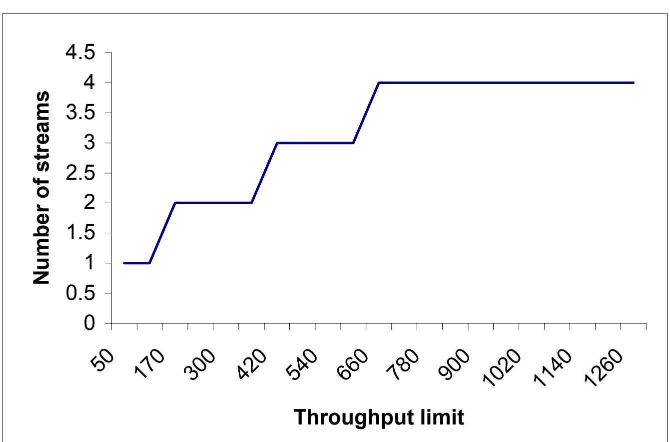


**Figure 3.** Server throughput limit vs. number of streams

## 5 Optimizing the Algorithm Using Receivers Partitioning

All adaptive multicast applications that require the server to solicit the receivers for feedback messages face the problem of feedback implosion, because the total number of feedback messages increases linearly with the number of receivers. Several approaches have been proposed to solve this problem including back-off timers [2, 3], probabilistic polling [4], and selecting group representative (or feedback aggregation) [5,16,17].

In our application, we avoided the implosion problem at two levels: first, we require that the receivers send their profiles and device limitations to the source only once, before the session starts, hence avoiding periodic feedbacks implosion. A problem with this approach is that it does not cover the variation over time of the bandwidth limit available for the receiver. To avoid this problem, users can use a conservative value for their bandwidth limit instead of the best all-time value.

To reduce further the implosion problem, we use class representation, where all receivers that have the same bandwidth limit are grouped together and are represented as one virtual receiver called the representative. The preferences of the representative are selected based on the preferences of the class members. The QoS selection algorithm would then select the QoS parameter for the class based on the QoS preferences of the representative. The decision to partition receivers according to their bandwidth limit is based on the fact that receivers that have the same bandwidth limit are more likely to have close preferences, and the values of their preferences is more likely to represent the preferences of individual receivers.

Even though receivers in the same class are more likely to have close preferences, there is still a range on the *minimum accepted* and *ideal* preference values for the class, and this gives several possibilities for selecting the preferences of the representative. Table 1 shows different variants for selecting the *minimum accepted* and *ideal* preference of the representative receiver based respectively on the *minimum accepted* and *ideal* value for all receivers in the class.

| | ***minimum accepted*** **value of the representative** | ***ideal*** **value of the representative** |
|---|---|---|
| **Variant 1** | Average of the *minimum accepted* values for all receivers | Average of the *ideal* values for all receivers |
| **Variant 2** | Minimum of the *minimum accepted* values for all receivers | Minimum of the *ideal* values for all receivers |
| **Variant 3** | Minimum of the *minimum accepted* values for all receivers | Maximum of the *ideal* values for all receivers |
| **Variant 4** | Maximum of the *minimum accepted* values for all receivers | Minimum of the *ideal* values for all receivers |
| **Variant 5** | Maximum of the *minimum accepted* values for all receivers | Maximum of the *ideal* values for all receivers |

**Table 1.** Variants of the preferences selection for the group representative

To evaluate the adequacy of the class representation, we selected one class of receivers, and run the selection algorithm once with all receivers directly considered by the source for the selection of the QoS parameters of the broadcast stream (no grouping) and another time with only the preferences of the class representative considered. We compared the average satisfaction of all the receivers in these two cases. Simulation results are shown in figure 4.

The graph in figure 4 shows clearly that variant 2 and 4 resulted in the worse average satisfaction for the group, even though the satisfaction of the representative with its selected parameters was one (1). This is basically due to the fact that the *ideal* preference for the group representative is the minimum of the *ideal* preferences of all receivers, reflecting hence the preferences of the most conservative receiver for the *ideal* preferences. The best variant for group representation is variant 5, where the *minimum accepted* value for the preferences of the representative is the maximum of the *minimum accepted* values for all receivers, and the *ideal* value for the representative is the maximum of all *ideal* values for all receivers. This variant avoids the conservative choice of the *minimum accepted* preference, and explores the optimism on the *ideal* preferences of all receivers.

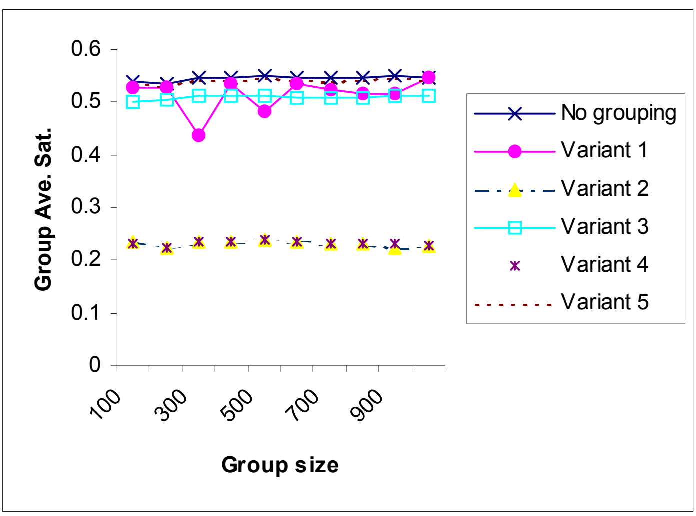


**Figure 4.** Average satisfaction with different variants of grouping.

## 6 Conclusion

In this paper, we have proposed an end-to-end rate-based mechanism for the selection of the optimum QoS for media broadcast that relies on the knowledge of the user preferences and bandwidth limitations of all receivers. Based on this information, the source will select the number of QoS variants for the given media stream and the QoS parameters for each of these variants. To limit the problem of feedback implosion from the receivers, we use a virtual representative for all receivers within a given group of receivers. Receivers are grouped according to the bandwidth limit of their device. We have considered several algorithms to determine the QoS preferences of the group representatives. Simulation results showed that not all of these algorithm lead to effective group representation. Simulation results also showed, as expected, that the average satisfaction of the receivers increases with an increase of the throughput of the source, which may limit the number and bandwidth of the available stream variants.

## Acknowledgements

The authors would like to thank Dr. Dwight Makaroff for his help and comments on this work.

## References

1. XU, X. R., Myers, A. C., Zhang, H. and Yavatkar, R., “Resilient multicast support for continuous-media applications,” *Proceedings of NOSSDAV'97.*
2. S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang, “A Reliable multicast framework for light-weight sessions and application level framing,” *in Proceeding of SIGCOMM*, 1995.
3. M. Grossglausser “Optimal deterministic timeouts for reliable scalable multicast,” IEEE Journal on Selected Area in Communications, vol. 15, no.3, 1997.
4. J. Bolot, T. Turletti, and I. Wakeman, “Scalable feedback control for multicast video distribution in the Internet,” in *Proc. Of ACM SIGCOMM*, pp. 58-67, August 1994.
5. R. Yavatkar, J. Griffoen, and M. Sudan, “ A reliable dissemination protocol for interactive collaborative applications,” in *Proceedings of INFOCOM*, 1991.
6. T. Jiang, M. Ammar, E. Zegura, “Inter-Receiver fairness: A Novel performance measure for Multicast ABR sessions,” in *Proceedings of ACM SIGMETRICS'98* , Madison, Wisconsin, June 1998.
7. T. Jiang, E. Zegura, M. Ammar, "Inter-Receiver Fair Multicast Communication over the Internet," in *Proc. of NOSSDAV 99*, June 1999.
8. S. McCanne, V. Jacobson, and M. Vetterli, “Receiver-driven layered multicast,” in *Proc. of ACM SIGCOMM*, pp. 117-130, August 1996.
9. L. Wu, R. Sharma and B. Smith, "Thin streams: An architecture for multicasting layered video," in *Proc. of NOSSDAV'97*, St.Louise,Missouri, May 1997.
10. S. Yan Cheung, Mostafa H. Ammar, and Xue Li , “On the use of destination set grouping to improve fairness in multicast video distribution,” in *Proc. of IEEE INFOCOM'96*, March 1996.
11. Brett J. Vickers, Célio Albuquerque and Tatsuya Suda, "Source-adaptive multi-layered multicast algorithms for real-time video distribution," in *IEEE/ACM Transactions on Networking*, 1999.
12. P. Assunção and M. Ghanbari, “Multi-Casting of MPEG-2 Video with Multiple Bandwidth Constraints,” in *Proc. of the 7$^{th}$ Int’l Workshop on Packet Video*, pp. 235-238, March 1996.

13. G. v. Bochmann and Z. Yang, ”Quality of service management for teleteaching applications using the MPEG-4/DMIF,” in *Intern. Workshop on Interactive Distr. Multimedia Systems and Telecom. Services*, Toulouse, Oct. 1999.
14. K. Elkhatib, G. v. Bochmann, “Selecting communication quality for mobile users based on device capabilities and user preferences,” Technical report. U of Ottawa, Canada, 2001.
15. A. Richards, G. Rogers, V. Witana, and M. Antoniades, “Mapping user level QoS from a single parameter,” in *Second IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, Versailles, November
16. Michael J. Donahoo and Sunila R. Ainapure, "Scalable multicast representative member selection", *INFOCOM 2001*.
17. D. DeLucia, K. Obraczka, “Multicast feedback suppression using representatives,” in *Proceedings of IEEE Infocom'97*.
18. K. El-Khatib, X. He, G. v. Bochmann, “*Quality of service negotiation based on device capabilities and user preferences*,” Technical report. U. of Ottawa, Canada. May 2000.