Single-loop packet merging for receiver-oriented multicast multi-layered video

Marcelo Dias de Amorim[1,2] Otto Carlos M. B. Duarte[1] Guy Pujolle[2]

[1]Grupo de Teleinformática e Automação – GTA
COPPE/EE – Universidade Federal do Rio de Janeiro
P.O. Box 68504 – 21945-970 – Rio de Janeiro – RJ – Brazil

[2]Laboratoire PRiSM
Université de Versailles St-Quentin-en-Yvelines
45, Avenue des Etats-Unis – 78035 Versailles – France

Abstract

For the transport of multicast video, the use of multi-layered coding is one of the most promising techniques. To avoid feedback implosion at the source due to the control packets returned by the receivers, the network has to implement packet merging at intermediate nodes. We propose in this paper the direct algorithm that performs the merging procedure in only one loop. This algorithm is optimized to achieve better video quality at heterogeneous receivers and results in less bandwidth requirements. We also propose the direct algorithm with virtual number of layers. In this case, the intermediate nodes consider that the source can transmit more than the actual number of layers. The results show that these algorithms lead to greater global "level of satisfaction" at the destinations.

1 Introduction

In multimedia communication, the transport of video flows has always been a subject of great interest. Many strategies have been proposed to solve problems related to the different characteristics of video traffics. The variable bit rate behavior of compressed video streams, for instance, introduces some constraints in the way bandwidth is allocated to traffics. One approach is to use resource reservation in order to to provide a certain level of guarantee to flows [1]. An efficient resource reservation, however, requires a previous knowledge of the traffic characteristics [2]. A more recent approach consists in giving to the application a certain level of consciousness on the network condition [3, 4, 5, 6, 7]. In [8], two strategies are proposed to dynamically control the throughput of a traffic source: the variable compression and the hierarchical coding. For the variable coding, the source supervises the available bandwidth and computes the minimum signal compression rate that should be applied in order to maximize the bandwidth utilization. For the hierarchical coding, each signal is coded in a base layer and in one or more improvement layers. These layers are related according to a causal law, where the quality of the presentation depends on the number of layers that are retrieved.

Many video applications are multicast by nature. Furthermore, in a multicast environment, it is likely that different receivers have different capacities in terms of available bandwidth. This heterogeneity turns difficult the task of choosing the adequate source's throughput. In a network with such characteristics, the utilization of multi-layered video can lead to a greater level of user's "satisfaction." Transmitting several video layers allows heterogeneous destinations to receive different global throughput.

The utilization of multi-layered video, however, introduces several challenges in the deployment of the communication system. The system must be able to handle the distribution of multiple related flows through a multicast communication. One approach is to use resource reservation to allocate bandwidth to each flow. A more robust solution consists in using feedback control to adjust video requirements to meet current network capabilities.

Previous works have proposed some schemes to deal with the real-time aspect of video distribution. In [9], the authors present their experience with control mechanisms for packet video in the Internet. The source transmits one video flow to multiple IP destinations. Receivers send regularly (to the source and to the other receivers) multicast reports containing, among other information, the quality of the video received. The source uses the information to estimate the average video quality and the congestion level of the network. Then, the source adjusts the sending rate to network conditions. Since this scheme uses only one video layer, overall level of user's satisfaction can be prejudiced by heterogeneity in the paths to the receivers.

In the destination set grouping [10], the source maintains a certain number of video flows derived from the same raw video. The receivers are classified according to their capabilities and each video flow is addressed to a particular group. Destinations that change their characteristics can move from one group to another in order to adapt their new capabilities to the most adequate destination group. Whereas this approach can lead to good fairness, it may be inefficient in terms of bandwidth utilization due to its redundancy of information.

The receiver-driven layered multicast [11] is a rate-adaptive protocol for the distribution of layered video using multiple IP-multicast groups. The source generates a certain number of video layers and transmits each layer in a different multicast session. The receivers subscribe to the number of groups they will. The communication is dynamic in the way receivers can dynamically join or leave groups. However, they are limited to the layers the source decides to transmit. Since the receivers do not have any influence on the way the rates of each layer are calculated, these values can be inappropriate to achieve good video quality at destinations and reasonable network utilization.

In [12], a number of feedback control packets containing current congestion state are exchanged between the source and the receivers. Based on the information, the source estimates the number of video layers and the respective rates. This mechanism is an extension of the work presented by the same authors in [13], in which two video layers are generated. To avoid implosion of feedback control packets at intermediate nodes, it is implemented an algorithm to merge feedback packets returned by the receivers.

In the literature, however, little attention has been given to the problem of merging feedback control packets in order to maximize the average level of satisfaction at the receivers. We propose in this paper the direct algorithm to efficiently merge feedback packets at intermediate nodes. The main goal is to calculate the throughput of each video layer in order to increase the global video quality at different destinations. The global video quality is estimated in terms of the level of satisfaction at each destination, i.e., the received video with regard to the maximum quantity that could be received without loss. Through the analysis of different network topologies, we prove that the proposed scheme leads to increased global quality of reception at heterogeneous receivers. Indeed, the obtained results show that the proposed mechanism results in less allocated bandwidth for greater level of user satisfaction. We also address in this paper the generation of feedback packets with a virtual number of layers. In this case, the intermediate nodes consider that the source can transmit more than its maximum capacity. We show that it results in an improvement in the average global video quality.

The following of this paper is organized in four sections. In section 2 we propose the single-loop algorithm. In section 3, the virtual layer technique is presented. Several analysis and simulations are described in section 4 and section 5 concludes this work.

2 Single-loop Merging Algorithm

We propose in this paper an efficient algorithm to merge feedback packets at intermediate nodes. This algorithm is also used to calculate the rates of the multiple layers at the source. For the exchange of control packets, we use a control scheme similar to the rate-based approach proposed in [12]. In this scheme, the source periodically multicasts feedback packets to destinations. As these feedback packets traverse the network, intermediate nodes mark them with the amount of available bandwidth. Each receiver copies the packet's contents into a return feedback packet and sends it to the source. Each feedback packet generated by a receiver must contain the maximum rate that this destination can receive. When these packets are sent to the source, intermediate nodes merge upcoming concurrent packets in order to avoid feedback implosion.

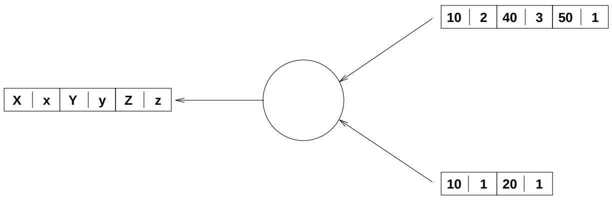


Figure 1: Merging feedback packets.

Figure 1 shows an example of the merging procedure at a node with two arriving links. Each feedback packet has at the most three entries, which means that the source must transmit at the most three video layers. Each entry contains two fields: one that stores a video rate and another with the number of destinations that want to receive this rate. In the same figure, the first entry of the packet that arrives in link 1 informs that two receivers will to receive 10Mbps. According to the second entry, three destinations want to receive 40Mbps. Finally, last entry states that one receiver wants to have the video at 60Mbps.

The algorithm to merge the packets performs as follows. The arriving feedback packets are stored in a temporary array with the video rates disposed in a crescent order. If this temporary array has a number of entries greater than the maximum number of layers that the source can generate, then the node performs an algorithm to reduce the number of entries. The algorithm discards the entry j that results in the larger "goodput rate" (G). The goodput rate is the total throughput received by the destinations without loss and is calculated as

$$G = \sum r_i \times c_i, \tag{1}$$

where r_i is the rate required by c_i receivers. The number of destinations in the discarded entry is added to the $j-1$th entry. These steps are repeated until the number of entries in the temporary array is reduced to the maximum number of layers that the source can generate.

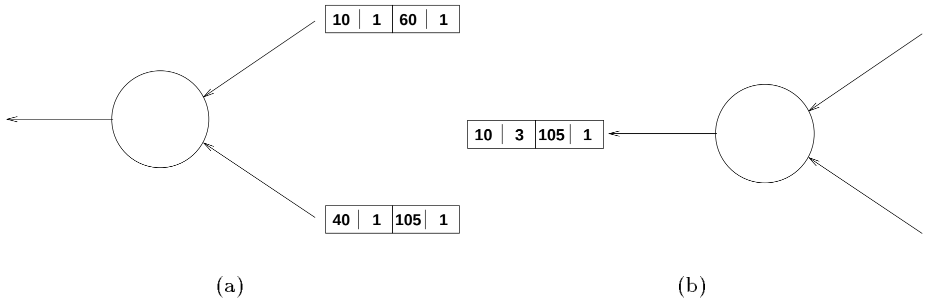


Figure 2: Examples of (a) arriving packets and (b) merged packet.

Although discarding the entry that results in the greater goodput rate, this algorithm does not optimize the global "level of satisfaction" at the receivers. For instance, suppose the circumstances shown in figure 2(a). Consider also G_j as the combined goodput rate when discarding entry E_j. In the first step of the merging procedure, we have four entries in the temporary array. Since the maximum number of video layers that the source can transmit is two, we must then discard two entries[1]. The sequence below shows each step of the reducing algorithm.

Step 1 $G_{40} = 10 \times 2 + 60 \times 1 + 105 \times 1 = 185$
$G_{60} = 10 \times 1 + 40 \times 2 + 105 \times 1 = 195$
$G_{105} = 10 \times 1 + 40 \times 1 + 60 \times 2 = 170$

Since G_{60} is the maximum goodput rate, this entry is eliminated and the counter of entry E_{60} is added to the counter of entry E_{40}.

Step 2 $G_{40} = 10 \times 3 + 105 \times 1 = 135$
$G_{105} = 10 \times 1 + 40 \times 3 = 130$

We finally discard E_{40} (step 2) and the node forwards the resulting feedback packet, as shown in figure 2(b). In conformity with this packet, the receivers in the sub-tree are able to receive two video layers, one at 10Mbps and the other at 95Mbps. However, these values are not optimal in the case of the example of figure 2. We consider the best result the one that leads to the greater level of user's satisfaction. In the example, there are four destinations in the sub-tree. The correspondent capabilities are r_1=10Mbps, r_2=40Mbps, r_3=60Mbps and r_4=105Mbps, respectively. If the source sends two layers at 10Mbps and 95Mbps, destinations D_1 and D_4 will be entirely satisfied, but D_2 and D_3 will not be served with their maximum capabilities. D_2 will have a degradation of 30Mbps (capacity=40Mbps, received video = 10Mbps) and D_3 a degradation of 50Mbps (capacity=60Mbps, received video = 10Mbps). Table 1 shows the level of global degradation for all possibilities of video rates. It should be emphasized that the smallest global degradation corresponds to the largest global video quality at the destinations.

[1] Since the first entry corresponds to the receivers with less capabilities, it is never discarded.

Table 1: Global average video degradation.

	$L_1 = 10; L_2 = 30$		$L_1 = 10; L_2 = 50$		$L_1 = 10; L_2 = 95$	
Destination	Received	Degradation	Received	Degradation	Received	Degradation
$D_1 = 10$	10	0	10	0	10	0
$D_2 = 40$	40	0	10	30	10	30
$D_3 = 60$	40	20	60	0	10	50
$D_4 = 105$	40	65	60	45	105	0
Global Degradation		85		75		80

We can see from table 1 that the best global video quality corresponds to the base layer at 10Mbps and the enhancement layer at 60Mbps. The algorithm to merge packets presented above, however, does not lead to the best values of video rates. This is because the algorithm performs the packet reduction in successive steps.

Consider that a node have to reduce a temporary packet which contains a fragment as shown in figure 3. In the picture, f_i is the required video rate and f_i^* corresponds to the number of destinations that want to receive this rate. Also consider that the fragment of figure 3 will be reduced as if it is an independent packet. This assumption is just to guarantee that no other entry would be discarded while this fragment is being reduced.

•••	f_{i-1}	f_{i-1}^*	f_i	f_i^*	f_{i+1}	f_{i+1}^*	f_{i+2}	f_{i+2}^*	•••

Figure 3: Temporary array.

Consider that the best values for the layers are f_{i-1} and f_{i+1}. The successive method does not lead to the optimized reduced feedback packet if

$$f_i^*(f_i - f_i) < f_{i+1}^*(f_{i+1} - f_i) + f_{i+2}^*(f_{i+1} - f_i) \tag{2}$$

and

$$f_{i+2}^*(f_{i+2} - f_{i+1}) < f_{i+1}^*(f_{i+1} - f_i) + f_{i+1}^*(f_i - f_{i-1}). \tag{3}$$

However, the successive method will not lead to this result if

$$f_i^*(f_i - f_i) > f_{i+1}^*(f_{i+1} - f_i) < f_{i+2}^*(f_{i+2} - fi + 1). \tag{4}$$

Combining equations (2), (3) and (4), we have that if

$$f_{i+1}^*(f_{i+1} - f_i) < f_i^*(f_i - f_i) < f_{i+1}^*(f_{i+1} - f_i) + f_{i+2}^*(f_{i+1} - f_i) \tag{5}$$

and

$$f_{i+1}^*(f_{i+1} - f_i) < f_{i+2}^*(f_{i+2} - f_{i+1}) < f_{i+1}^*(f_{i+1} - f_i) + f_{i+1}^*(f_i - f_{i-1}) \tag{6}$$

are satisfied, then the global video quality is degraded if we use the successive method.

To avoid the situations as in the example above, this paper proposes the Direct Algorithm to merge feedback packets. This algorithm computes the final merged feedback packet in only one step. In a general case where the source can generate L video layers ($C_1 ... C_L$) and the packet that must be reduced has N entries, the direct algorithm performs as follows. We make $C_1 = f_1$ and we calculate $C_2 ... C_L$ so as to minimize the *loss sum* (L_S), calculated as

$$L_S = \sum_{i=2}^{L} f_i^* \times (f_i - r_k), \tag{7}$$

where k is the maximum number of layers that the rate f_i can support[2], and r_k is calculated as

[2]For example, if $f_i = 100$ and we have three layers at 50, 40 and 30, then $k = 2$, because the receiver is able to receive only the first two layers.

$$r_k = \sum_{j=1}^{k} C_j. \tag{8}$$

The values of $C_2...C_L$ are calculated in order to coincide with a combination of f_j. Furthermore, determining $min\{L_S\}$ we guarantee that the global average video quality is locally optimized at the intermediate nodes.

3 Virtual Layers

The procedure of merging feedback packets may prevent the source to choose the best values for the video layers. Depending on the number of intermediate nodes that perform the merging algorithm, entries that should be chosen as actual video layers may be eliminated before achieving the source.

Consider the situation depicted in figure 4. In this example, the source can transmit up to two video layers. In node N_2, the entry that corresponds to the destination that wants to receive 50Mbps is eliminated, and this information is lost. If this information had not been lost, the algorithm would find the rates that lead to the best global video quality, i.e., 10Mbps and 50Mbps. In this case, the global degradation would be 80Mbps. When performing the packet merging at N_1, however, the obtained throughputs are 10Mbps and 100Mbps, which results in global degradation of 100Mbps.

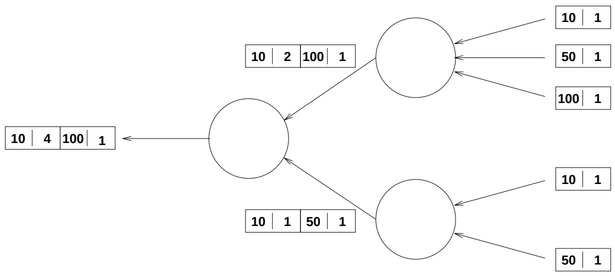


Figure 4: Information loss when merging packets.

To minimize this problem, we also propose in this paper the merging algorithm with virtual layers. In this algorithm, the intermediate nodes consider that the source can transmit more than the actual number of layers. This strategy may avoid information loss as in the example of figure 4. The intermediate nodes are informed that the source can transmit V layers, with $V > C$ (C is the actual number of layers transmitted by the source.)

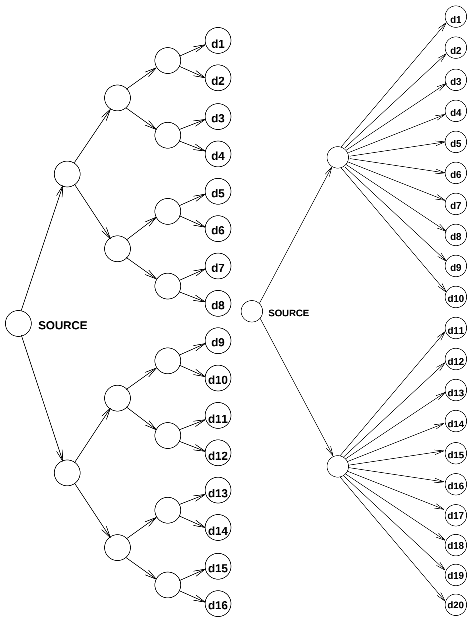
In the same example above, consider that $V = 3$. The merging algorithm does not perform at node N_2. Consequently, the entry that corresponds to the receivers that want to receive 50Mbps is not discarded. At node N_1, this information is still existing. When performing the merging algorithm, node N_1 discards the entry that stores 100Mbps, which leads to an optimized merging procedure.

4 Analysis

To evaluate the direct algorithm, we have performed several simulations upon different scenarios. Each scenario corresponds to a multicast video session with one sender and several receivers. The analysis is based on the global video degradation and on the network bandwidth utilization. We consider in our simulations a source with different capabilities, i.e., with different values for the maximum number of layers. The main goals are to evaluate the tradeoff between the number of layers and the level of satisfaction and to analyze the introduction of virtual layers at the intermediate nodes.

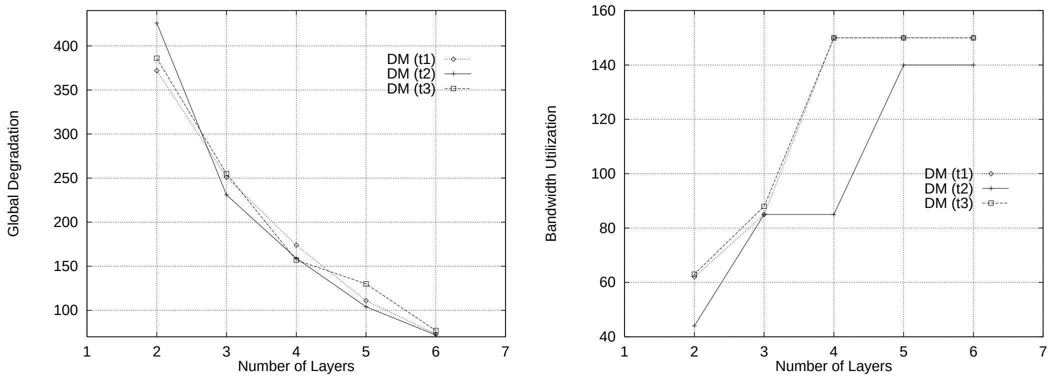
The first topology is shown in figure 5(a). This topology represents a network with the receivers sparely distributed among the branches. We have also performed simulations for a multicast session with concentrated receivers (figure 5(b)). The receivers' capabilities were randomly generated with average rate values between 10 and 150.

Figure 6(a) shows the global video quality degradation for the direct method. The curves correspond to the results for three different merging loops at $t = t_0$, $t = t_1$ and $t = t_2$. The correspondent levels of bandwidth utilization are shown in figure 6(b). We observe from these two graphics that, for example, at $t = t_2$, the bandwidth utilization is the same if the source transmits 3 or 4 layers. But transmitting 4 layers, the global degradation is approximately 30% less than if the source transmits 3 layers.



(a) (b)

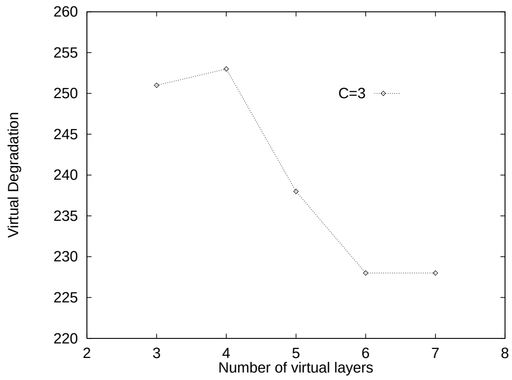
Figure 5: Simulation topologies: (a) uniformly distributed and (b) concentrated.



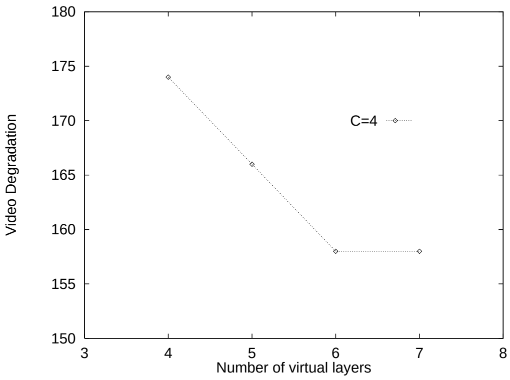
(a) (b)

Figure 6: Topology 1: (a) global degradation and (b) bandwidth utilization.

In figure 7, the direct method is evaluated for different numbers of virtual layers. In this example, the source is supposed to transmit up to 3 video layers for figure 7(a) and up to 4 video layers for figure 7(b). In both cases, the curves saturate for a number of virtual layers greater than 6, which corresponds to the maximum average level of satisfaction at the receivers.



(a)



(b)

Figure 7: Source transmitting (a) 3 layers and (b) 4 layers.

To compare the direct and the successive method, we have simulated the two topologies for the number of video layers varying from 2 to 6. Table 2 shows some results obtained for several values of C. For the direct method, the number of virtual layers is set to $L = C + 3$. Observe that the direct algorithm results in better global video quality without requiring additional bandwidth.

Table 2: Global average video degradation.

Topology 1

	Direct Method		Successive Method	
C	Degradation	Utilization	Degradation	Utilization
2	361	71	413	85
3	228	80	233	80
4	158	150	174	150
5	111	150	111	150

Topology 2

	Direct Method		Successive Method	
C	Degradation	Utilization	Degradation	Utilization
2	539	75	539	75
3	339	115	421	115
4	223	115	223	115
5	163	143	188	143

5 Conclusions

In this paper, we have addressed the problem of managing control information for multicast distribution of multi-layered video. In this paper, we proposed the direct algorithm to merge feedback packets at intermediate nodes. Contrary to other proposals found in the literature, this algorithm performs the merging procedure in only one loop. This leads to improved global level of satisfaction at heterogeneous destinations. The results show that the proposed method achieves lower degradation without additional allocated bandwidth.

We have also proposed the direct method with virtual number of layers, i.e., the intermediate nodes consider that the source can transmit a number of layers greater than its maximum capacity. As the simulations show, it improves global video quality since merging procedures at some intermediate nodes are no longer performed. In some cases, the algorithm achieves the same levels of satisfaction as if there were no packet merging.

References

[1] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource reservation protocol," *IEEE Network Magazine*, Sept. 1993.

[2] M. D. de Amorim and O. C. M. B. Duarte, "A novel deterministic traffic model based on a two-level analysis of MPEG video sources," in *IEEE GLOBECOM'98*, (Sydney, Australia), pp. 244–249, Nov. 1998.

[3] D. Sisalem and A. Wolisz, "Towards TCP-friendly adaptive multimedia applications based on RTP," in *Fourth IEEE Symposium on Computers and Communications (ISCC'99)*, (Red Sea, Egypt), July 1999.

[4] D. G. Wadington and D. Hutchison, "A general model for QoS adaptation," in *IEEE/IFIP International Workshop on Quality of Service (IWQoS)*, (Napa, CA, USA), May 1998.

[5] B. Li and K. Nahrstedt, "A control theoretical model for quality of service adaptations," in *IEEE/IFIP International Workshop on Quality of Service (IWQoS)*, (Napa, CA, USA), May 1998.

[6] T. Gross and P. Steenkiste, "A perspective on application/networking coupling," in *8^{th} International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, (Cambridge, UK), July 1998.

[7] D. Sisalem, "Fairness of adaptive multimedia applications," in *IEEE ICC'98*, (Atlanta, GA), June 1998.

[8] C. Diot, C. Huitema, and T. Turletti, "Multimedia applications should be adaptive," in *HPCS'95*, (Mystic, CN), Aug. 1995.

[9] J. C. Bolot and T. Turletti, "Experience with control mechanisms for packet video in the internet," *ACM Computer Communication Review*, vol. 28, Jan. 1998.

[10] S. Y. Cheung, M. H. Ammar, and X. Li, "On the use of destination set grouping to improve fairness in multicast video distribution," in *IEEE Infocom*, (San Francisco, CA, USA), Mar. 1996.

[11] S. McCanne, V. Jacobson, and M. Vitterli, "Receiver-driven layered multicast," in *ACM SIGCOMM*, (Stanford, CA, USA), Aug. 1996.

[12] B. J. Vickers, C. Albuquerque, and T. Suda, "Adaptive multicast of multi-layered video: Rate-based and credit-based approaches," in *IEEE Infocom*, (San Francisco, CA, USA), Mar. 1998.

[13] B. J. Vickers, M. Lee, and T. Suda, "Feedback control mechanisms for real-time multipoint video services," *IEEE Journal on Selected Areas in Communications*, vol. 15, Apr. 1997.