Improving User Satisfaction in Adaptive Multicast Video

Marcelo Dias de Amorim, Otto Carlos M. B. Duarte, and Guy Pujolle

Abstract: **Adaptability is the most promising feature to be applied in future robust multimedia applications. In this paper, we propose the Direct Algorithm to improve the degree of satisfaction at heterogeneous receivers in multi-layered multicast video environments. The algorithm relies on a mechanism that dynamically controls the rates of the video layers and is based on feedback control packets sent by the receivers. The algorithm also addresses scalability issues by implementing a merging procedure at intermediate nodes in order to avoid packet implosion at the source in the case of large multicast groups. The proposed scheme is optimized to achieve high global video quality and reduced bandwidth requirements. We also propose the Direct Algorithm with a virtual number of layers. The virtual layering scheme induces intermediate nodes to keep extra states of the multicast session, which reduces the video degradation for all the receivers. The results show that the proposed scheme leads to improved global video quality at heterogeneous receivers with no cost of extra bandwidth.**

Index Terms: **Adaptive applications, multi-layering, multicast, virtual layering, video quality.**

I. INTRODUCTION

Heterogeneity in intermediate and terminal equipment is a major problem for the deployment of multicast applications in current network architectures such as the Internet. On the one hand, if the source rate is compatible with the slowest receivers, the paths to the faster receivers can be underutilized. On the other hand, if the source sends at higher rates, slower receivers will not be able to receive data. These challenges are intrinsic to video distribution, since most video applications are multicast by nature [1]–[3].

We propose in this paper an adaptive system for improving the global quality of the multicast session in networks with heterogeneous receivers. The system is based on cooperative source-receivers and uses the multi-layered approach for differentiating groups of receivers with similar capacities [4]–[7]. Source and receivers exchange control packets containing information about network states. Based on these packets, the source then adapts the rates of the video layers to current network conditions. If all packets sent by the receivers arrive at the source, a feedback implosion occurs and the system suffers from a source collapse [8]–[11].

Manuscript received April 27, 2000; approved for publication by Alex Gelman, Division III editor, February 28, 2002.

M. D. de Amorim and G. Pujolle are with the Network and Performance Group of the LIP6 Laboratory, University of Paris VI, France, e-mail: {Marcelo.Amorim,Guy.Pujolle}@lip6.fr.

O. C. M. B. Duarte is with the Network and Automation Group (GTA/COPPE), Federal University of Rio de Janeiro, Brazil, e-mail: otto@gta.ufrj.br.

This work has been supported by CAPES, COFECUB, CNPq, UFRJ, UPMC, RNRT, and CNRS.

Previous works have proposed some adaptive schemes to deal with the real-time aspect of video distribution [1], [12]–[18]. In [1] and [14], a number of feedback control packets carrying current congestion state are exchanged between the source and the receivers. Based on this information, the source estimates the number of video layers and the respective rates. To avoid implosion of feedback control packets at intermediate nodes, an algorithm is implemented to merge feedback packets returned by the receivers. In [16], the source transmits one video flow to multiple IP destinations. Based on reports sent by the receivers, the source estimates the average video quality and the congestion level of the network to adjust the video rate. Since this scheme uses only one video layer, global video quality may be degraded due to heterogeneity in the receivers. In the destination set grouping [17], 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. Whereas this approach can lead to good fairness, it may be inefficient in terms of bandwidth utilization due to redundancy of information. In the receiver-driven layered multicast [18], the source generates a certain number of video layers and transmits each layer to a different IP-multicast group. The receivers subscribe to a subset of the groups, and this subset depends on the capacity of the receivers. In this approach, the communication is dynamic because the receivers can dynamically join or leave groups. Nevertheless, they are limited to the layers the source decides to transmit.

The core of our system is the Direct Algorithm that aims at eliminating the problem of feedback implosion by implementing a procedure at intermediate nodes that merges concurrent control packets. The algorithm is implemented in order to maximize the global video quality at the receivers. The global video quality is estimated in terms of the level of satisfaction at each destination, i.e., the difference between the required and received video rates. Through the analysis of different network topologies, we prove that the proposed scheme leads to improved global quality of reception at heterogeneous receivers. Indeed, the results show that the proposed mechanism results in less allocated bandwidth for higher levels of user satisfaction.

We also address in this paper the Direct Algorithm using the concept of virtual layering, where intermediate nodes consider that the source is able to transmit more layers than its actual capacity. We show that this feature avoids discarding relevant information that should be stored for further comparison with control information of neighbor subnetworks. This means that the virtual layering scheme reduces the probability of eliminating important control information during a merging procedure. Our analysis show that this approach leads to results that are close to the empirical border, which is obtained when no merging procedures are performed at intermediate nodes. In this particular case, since every information sent by the receivers arrives at the source, this latter can compute the video layers in such a way to obtain the highest possible global video quality.

Through the analysis of different network topologies, we show that the proposed scheme leads to improved global quality of reception at heterogeneous receivers. Indeed, the results show that the proposed mechanism results in less allocated bandwidth for higher levels of user satisfaction.

The remainder of this paper is organized as follows. In Section II, we propose and analyze the Direct Algorithm. Section III presents the Direct Algorithm with virtual layering. Several analysis methods and simulations results are described in Section IV. Finally, Section V concludes this paper.

II. IMPROVING SATISFACTION AT RECEIVERS

The adaptive system proposed in this paper consists of a multi-layered multicast session where the rates of the video layers are computed based on feedback control packets sent by the receivers. The main component of the system is the algorithm used to merge the control packets in order to avoid feedback implosion at the source. In this scheme, the source periodically multicasts control packets (called inspection packets) to the destinations. As these control packets traverse the network, intermediate nodes mark them with the amount of available bandwidth in the links, $c(l)$, where l is a link in the multicast tree. When a receiver r, belonging to the multicast session m, receives this control packet, it computes the maximum rate it can receive, given by

$$b^{r,m} = \min_{l \in L_{s,r}} c(l), \tag{1}$$

where $L_{s,r}$ is the set of links traversed by the control packet from the source s to the receiver r. The receiver then builds a return feedback packet that contains this rate and sends it back to the source. When routing the packets from the different receivers, intermediate nodes merge them according to a merging algorithm in order to avoid feedback packet implosion. We note that in order to have the appropriate functioning of the system, the path from the source to the receivers and from the receivers to the source should be the same, which can be easily implemented by efficient routing algorithms like HBH [19] or MOSPF [20].

Let us now fix our targets. For the merging procedure, we define the fairest output feedback control packet to be the one that satisfies the following definitions.

Definition 1: The global video degradation for a multicast multi-layered session is the sum of the differences between the required video rate and the received video rate for every receiver taking part in the session.

Definition 2: After a merging procedure, the fairest output feedback control packet is the one that leads to the lowest global video degradation at the receivers.

A feedback control packet has k entries, $1 \leq k \leq L$, where L is the number of video layers transmitted by the source. Each entry $e_i, i = 1, \cdots, k$, has two fields: a video rate f_{e_i}, and the number of receivers $f^*_{e_i}$ that require the rate f_{e_i}. Our system is similar to the control scheme proposed in [14] (we will call this scheme the "classic approach"), where the arriving feedback packets are stored in a temporary array $\vec{t}$ for further processing. The merging procedure is executed when a timer T, set upon the arrival of the first feedback packet, goes off. This allows packets to be accumulated and avoids unbounded waiting if expected packets do not arrive. The entries of the temporary array are organized in such a way that $f_{e_{i-1}} \leq f_{e_i} \leq f_{e_{i+1}}, \forall i$. If $f_{e_i} = f_{e_{i+1}}$, then e_i and e_{i+1} are combined by performing[1]

$$\begin{aligned} f_{e_i,\oplus} &= f_{e_i,\ominus}, \\ f^*_{e_i,\oplus} &= f^*_{e_i,\ominus} + f^*_{e_{i+1},\ominus}, \end{aligned} \tag{2}$$

where the symbols $\ominus$ and $\oplus$ represent the values of the variables respectively before and after the merging procedure. If the temporary array has a number of entries c_T greater than the number of layers transmitted by the source, then the node performs the algorithm to eliminate $c_T - L$ entries. We call a "candidate packet" any packet that can be obtained by the merging procedure. The candidate packet that will be chosen to be forwarded to the upstream node depends on the policy applied by the merging algorithm.

The classic approach discards the entry j that leads to the highest goodput rate G. The goodput rate is the total rate received by the receivers when discarding entry e_j and is computed as

$$G = \sum_{i=1, i \neq j}^{c_T} f_{e_i} f^*_{e_i}. \tag{3}$$

The number of destinations $f^*_{e_j}$ in the discarded entry is added to the number of destinations in the $(j-1)$th entry. These steps are performed in a loop until the number of entries in the temporary array is reduced to the maximum number of layers transmitted by the source.

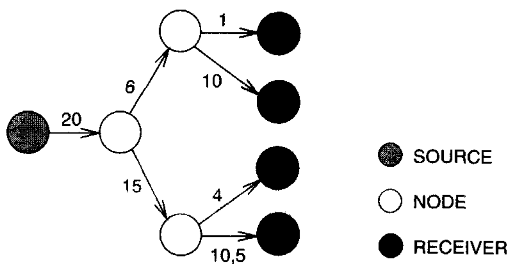
Although discarding entries that lead to the highest goodput rate, this approach does not guarantee that the quality of the video at the receivers will be optimized. Suppose the circumstances shown in Fig. 1(a) where the source transmits two video layers. Consider also G_{e_j} to be the goodput rate when discarding entry e_j. At the node that precedes the source, the temporary array has four entries, then two of them must be discarded. We have decided not to discard the first entry in order to provide every receiver with at least the base layer. The sequence below and Fig. 1(b) show each step of the reducing algorithm.[2]

Step 1 $\vec{t} = \{(1,1), (4,1), (6,1), (10.5,1)\}$
$G_{e_2} = 18,5$
$G_{e_3} = 19,5$
$G_{e_4} = 17.$

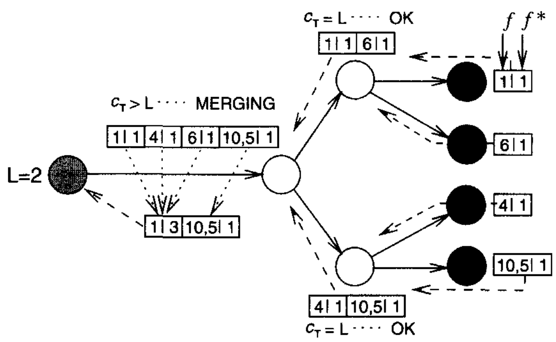
G_{e_3} is the maximum goodput rate, then entry e_3 is discarded and $f^*_{2,\oplus} = f^*_{2,\ominus} + f^*_{3,\ominus}$. Immediately before discarding the second entry, the temporary array has the following three entries

[1] In practice, two entries are automatically merged if the difference $f_{e_{i+1}} - f_{e_i}$ is within a certain threshold α. In this case, we say that these entries are "compatible."

[2] For simplicity reasons, in some examples throughout this paper we will abuse and omit sometimes the units of the variables.



(a) Network state.



(b) Merging procedure.

Fig. 1. Evolution of the merging procedure.

$$\begin{array}{lll} e_{1,\ominus} \Rightarrow & f_{1,\ominus} = 1 & f^*_{1,\ominus} = 1 \\ e_{2,\ominus} \Rightarrow & f_{2,\ominus} = 4 & f^*_{2,\ominus} = 2 \\ e_{3,\ominus} \Rightarrow & f_{3,\ominus} = 10,5 & f^*_{3,\ominus} = 1. \end{array}$$

Step 2 then follows.

$$\textbf{Step 2} \quad \begin{array}{l} \vec{t} = \{(1,1),(4,2),(10.5,1)\} \\ G_{e_2} = 13,5 \\ G_{e_3} = 13. \end{array}$$

In step 2, G_{e_2} is greater than G_{e_3}. Thus, e_2 is discarded and the node forwards the resulting feedback packet. The procedure is illustrated in Fig. 1(b). In conformity with the resulting feedback control packet, receivers in the sub-tree will be able to receive two video layers, one at 1 and the other at 9,5.[3] Nevertheless, these rates are not optimal because they do not lead to the highest level of user satisfaction. In fact, the rates that would result in the highest level of user satisfaction at the receivers are $l_1 = 1$ and $l_2 = 5$. We show in the following why the classic approach presented above does not lead to the best results.

Let $\mathbf{S}$ be a subset of the entries in the temporary array and the number of elements in $\mathbf{S}$ be $|\mathbf{S}| = c_T - L$. $\delta_{\mathbf{S}}$ is the global video degradation when discarding entries $e_i \in \mathbf{S}$. b_{r_i} is the video rate required by receiver r_i and $\hat{b}_{r_i}$ is the received rate. δ_{r_i} is the degradation at receiver r_i, i.e., $\delta_{r_i} = b_{r_i} - \hat{b}_{r_i}$. Consequently, we have that

$$\delta_{\mathbf{S}} = \sum_{i=1}^{N} \delta_{r_i}, \tag{4}$$

where N is the total number of receivers in the multicast session. In the example of Fig. 1, we then have

$$\begin{aligned} \delta_{(e_2,e_3)} &= 8,5, \\ \delta_{(e_2,e_4)} &= 7,5, \\ \delta_{(e_3,e_4)} &= 8. \end{aligned}$$

Recall that the lowest global degradation corresponds to the highest global video quality at the destinations, which is in the above example $\delta_{(e_2,e_4)} = 7,5$. This corresponds to the base layer $l_1 = 1$ and the enhancement layer $l_2 = 5$. We show in the following that the classic algorithm does not lead to the highest global video quality because it performs the packet merging procedure in successive independent steps.

Let $\mathbf{U}$ be all possible output packets from a merging procedure. Let also $\mathbf{O} \subset \mathbf{U}$, $|\mathbf{O}| = L$, be the subset of entries that lead to the lowest global video degradation, and

$$\vec{t} = \{(f_{o_1}, f^*_{o_1}), \cdots, (f_{o_2}, f^*_{o_2}), \cdots, (f_{o_L}, f^*_{o_L}), \cdots, (f_{c_T}, f^*_{c_T})\}$$

be the temporary array to be reduced. Consider that $\hat{\mathbf{O}} \subset \mathbf{U}$, $|\hat{\mathbf{O}}| = L$, is a subset differing from $\mathbf{O}$ in at least one of the elements. The following inequality then holds:

$$\sum_{\forall k, k \notin \mathbf{O}} f^*_k (f_k - f_{\mathbf{O}_x}) < \sum_{\forall k, k \notin \hat{\mathbf{O}}} f^*_k (f_k - f_{\hat{\mathbf{O}}_x}), \quad \forall \hat{\mathbf{O}}, \tag{5}$$

where $\mathbf{O}_x$ is chosen in such a way that $\mathbf{O}_x \leq f_k < \mathbf{O}_{x+1}$. Nevertheless, the classic approach does not lead to this result if during *any* of the steps there is at least one entry $e_v \in \mathbf{O}$ such that

$$f^*_{e_v}(f_{e_v} - f_{e_{v-1}}) < f^*_{e_k}(f_{e_k} - f_{e_{k-1}}), \quad \forall k, k \neq v. \tag{6}$$

Let us perform such analysis in the case where the temporary array has four entries and two of them must be discarded. We then have

$$\mathbf{S}_T = \{(f_{e_1}, f^*_{e_1}), (f_{e_2}, f^*_{e_2}), (f_{e_3}, f^*_{e_3}), (f_{e_4}, f^*_{e_4})\}. \tag{7}$$

Supposing the fairest entries to be e_1 and e_3, the following inequalities must hold:

$$\begin{cases} f^*_{e_2}(f_{e_2} - f_{e_1}) < f^*_{e_3}(f_{e_3} - f_{e_2}) + f^*_{e_4}(f_{e_3} - f_{e_2}), \\ f^*_{e_4}(f_{e_4} - f_{e_3}) < f^*_{e_3}(f_{e_3} - f_{e_2}) + f^*_{e_3}(f_{e_2} - f_{e_1}), \end{cases} \tag{8}$$

By combining (5) and (8), and after some manipulation, we have that the classic approach will not lead to the fairest output packet if

[3] The total bandwidth the receivers can have are 1 and 10.5. This corresponds to the base layer at 1 and the enhancement layer at 9.5.

$$\begin{cases} f^*_{e_3}(f_{e_3}-f_{e_2}) < f^*_{e_2}(f_{e_2}-f_{e_1}) < \\ \quad f^*_{e_3}(f_{e_3}-f_{e_2}) + f^*_{e_4}(f_{e_3}-f_{e_2}), \\ \\ f^*_{e_3}(f_{e_3}-f_{e_2}) < f^*_{e_4}(f_{e_4}-f_{e_3}) < \\ \quad f^*_{e_3}(f_{e_3}-f_{e_2}) + f^*_{e_3}(f_{e_2}-f_{e_1}). \end{cases} \tag{9}$$

The Direct Algorithm proposed in this paper avoids the situations described above, and *always* leads to the fairest output packet at each node by performing the merging procedure in only one loop. In a general case where the source transmits L video layers $(l_1, \cdots, l_L)$ and at some intermediate node the packet that must be reduced has c_T entries, the Direct Algorithm works as follows. We make $l_1 = f_{e_1}$ and we compute $l_2, \cdots, l_L$ so as to minimize the *loss sum* Δ_S:

$$\Delta_S = \sum_{\forall k, k \notin \mathbf{O}} f^*_k(f_k - f_{\mathbf{O}_x}). \tag{10}$$

In the Direct Algorithm, all exceeding entries are discarded in the same loop, because all candidate packets are examined. Furthermore, by computing the minimum Δ_S we guarantee that the global average video quality is locally optimized at the intermediate nodes. We show below a pseudo-code of the Direct Algorithm.

```
Direct Algorithm (in: temporary array;
                  out: fairest output packet)
δ = 0;
δ_tmp = 0;
Compute O^tmp ← "first candidate packet";
while (∃ "next candidate packet")
{
   for (k = 1;k ≤ c_T;k + +)
      δ_tmp = δ_tmp + f*_k(f_k − f_O_x);
   if (δ_tmp < δ) then
   {
      δ = δ_tmp;
      O = O^tmp;
   }
   Compute next O^tmp ← "candidate packet";
}
```

One might ask about the complexity of the proposed algorithm since for each merging procedure the node must consider all possibilities for the discarded entries. We argue in the following that this is not always true. Consider a node n that is going to perform a merging procedure. This node has k incoming links $p^n_i, i = 1, \cdots, k$, and one output link p^n_{out}. We perform the analysis in the extreme case where the processing overhead at node n is maximum in order to define bounds on the performance of the system. For such case, we make the following assumptions:

- All incoming links $p^n_i, i = 1, \cdots, k$, belong to the multicast tree and have consequently in some downstream node at least one receiver sending feedback control packets.
- At the moment of the merging procedure, one feedback control packet has arrived at each incoming link. This means that all packets arrive at the node within an interval T equal to the interval size of the timer set upon the arrival of the first feedback packet. In other words, if the first packet arrives at t_0 and the last packet at t_{last}, then the inequality $t_0 + T > t_{last}$ must hold.
- All incoming packets have the maximum number of entries L equal to the number of video layers transmitted by the source. We also consider that the entries of all packets are "non-compatible", i.e., they do not have rates sufficiently close to one another to be automatically combined.

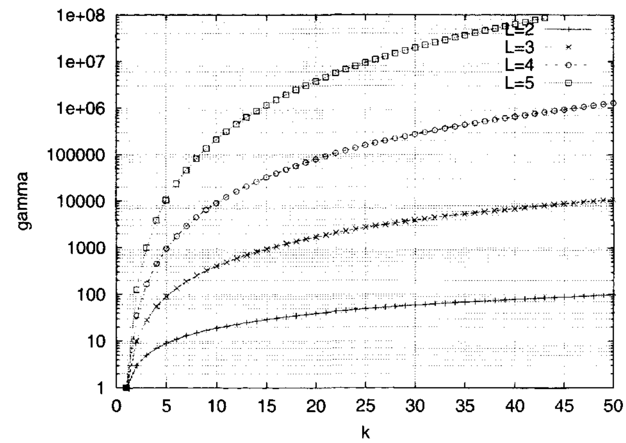


Fig. 2. How γ responds to k and L.

The first task of the merging algorithm is to build the temporary array by concatenating (and organizing) the incoming feedback packets. After performing this step, the number of entries in the temporary array is

$$\hat{n}_e = kL. \tag{11}$$

Candidate outgoing packets are obtained when $\hat{n}_e - L$ entries are discarded from the temporary array. The number of candidate packets γ is given then by

$$\gamma = \frac{\prod_{i=1}^{kL+1}(kL-i)}{\prod_{i=1}^{L+1}(L-i)\prod_{i=L}^{kL+1}(kL-i)} = \frac{\prod_{i=1}^{L-1}(kL-i)}{\prod_{i=1}^{L+1}(L-i)}. \tag{12}$$

Fig. 2 shows the behavior of γ for different values of L and k. We can observe that beyond a certain threshold the influence of L and k on the number of candidate packets γ is quite important. Nevertheless, in practical situations k and L are relatively small. If we consider for instance a source that has an embedded MPEG 2 encoder, the number of video layers may be up to three. If we look again at the curve of Fig. 2, for $L = 3$ we note that γ stays within a limited range even for large k. For $k = 15$ (i.e., a node must have 15 branches of the multicast tree and each branch must have an incoming packet of length L), the number of candidate packets is 1000. Furthermore, the processing of the output feedback control packet requires light computation since the algorithm uses only additive and comparative operations.

Nevertheless, if the number of candidate packets seems to be high, we can use an hybrid approach performed in two steps. During the first one, the merging procedure uses the classic approach to reduce the number of entries of the temporary array to a previously defined threshold. When this threshold is reached, the merging procedure is switched to the Direct Algorithm. This

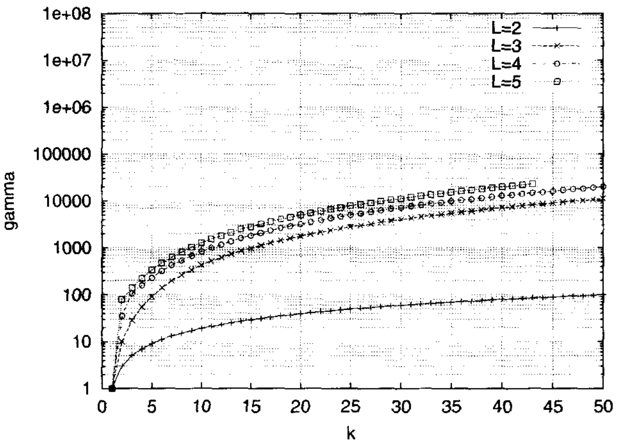


Fig. 3. How γ responds to k and L when the hybrid approach is used.

keeps the complexity of the system under control and allows the merging procedure to improve the quality of the multicast session. Fig. 3 shows the number of candidate packets for a range of values of L and k. We have chosen the threshold in such a way as to limit the number of candidate packets for the single-loop algorithm to $\beta_{max} = 100$. When compared to the curves of Fig. 2, we note that the number of candidate packets can be easily controlled by applying the hybrid approach.

III. VIRTUAL LAYERING

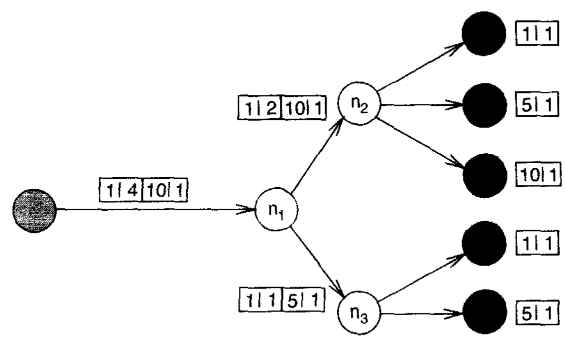
Depending on the number of intermediate nodes that perform the merging algorithm, entries that improve the global video quality may be inappropriately discarded. When performing a merging procedure, intermediate nodes handle only information sent by the receivers in their respective downstream sub-trees. Receivers in other sub-trees have no influence on the candidate output packets, and this is the price for reducing the signaling in the network.

In such non-cooperative scenario, the virtual layering scheme induces intermediate nodes to conserve extra entries in the feedback control packets by setting the number of layers transmitted by the source to a value $\lambda > L$. Thus, when performing a merging procedure, intermediate nodes reduce the size of the temporary array from c_T to λ. We will see in the following how the virtual layering scheme can reduce the limitations of non-cooperative sub-trees.

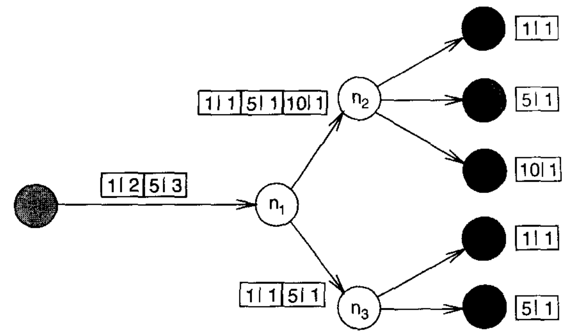
Consider the general network fragment where g nodes, $\mathbf{N} = \{n_1, n_2, \cdots, n_g\}$ are connected to node n_0 and there are k_{n_j} upcoming packets arriving at node $n_j \in \mathbf{N}$. Let $\beta_{p_i^{n_j}}$ be the number of entries in the ith packet $p_i^{n_j}$ and suppose that a merging procedure is performed in each node, i.e.:

$$\sum_{i=1}^{k_{n_j}} \beta_{p_i^{n_j}} > L, \qquad \forall j. \tag{13}$$

δ_{n_0} is the global video degradation in node n_0 when the fairest output feedback packet for node $n_j, j = 1, \cdots, g$, is $\mathbf{O}_{n_j}$. It is possible that $\exists \hat{\mathbf{O}}_{n_i}, j = 1, \cdots, g$, $\mathbf{O}_{n_i} \neq \hat{\delta}_{n_i}$, such that after applying the virtual layering scheme we have $\hat{\delta}_{n_0} < \delta_{n_0}$.



(a) Without virtual layering.



(b) Using one virtual layer.

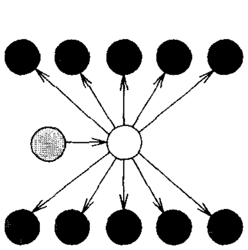
Fig. 4. Evolution of the merging procedure.

Consider the situation depicted in Fig. 4, where the source transmits two video layers. In the normal case, when performing the merging procedure, intermediate node n_2 discards $c_T - L = 1$ entry, which corresponds in our example to $e_2 = (5, 1)$. Naturally, this information is lost and cannot be recovered at the upstream nodes. Observe that the packet in the subtree that begins at node n_3 also contains an entry with the same rate $f = 5$. If the same entry were not eliminated at node n_2, the algorithm at node n_1 would not discard $f = 5$ but $f = 10$.

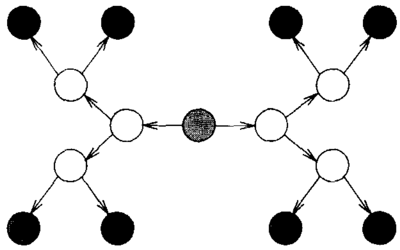
The virtual layering scheme reduces the probability of this type of problems occurring. In the same example above, if we do $\lambda = 3$, the merging algorithm is not performed at node n_2. Consequently, the entry that corresponds to the receivers that want to receive $f = 5$ is not discarded. At node n_1, this information still exists. When performing the merging algorithm, node n_1 discards the entry that stores $f = 10$, which leads to the highest global video quality.

IV. ANALYSIS

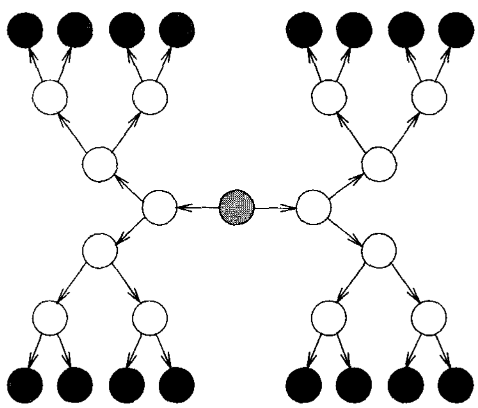
In this section, we analyze the Direct Algorithm and compare its performance with the classic approach using the NS network simulator [21]. We evaluate the level of video degradation and



(a) Topology 1.



(b) Topology 2.



(c) Topology 3.

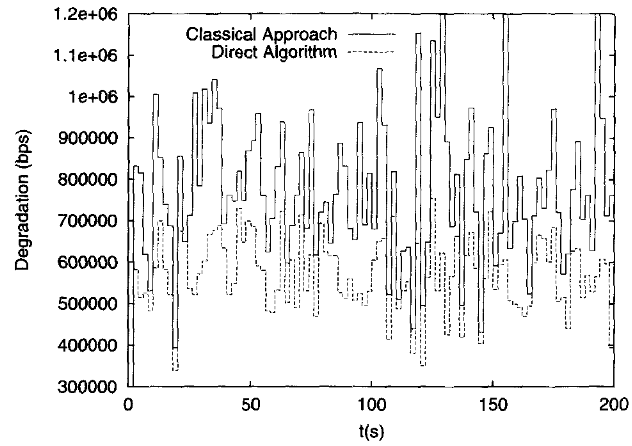
Fig. 5. Simulation topologies.

the network bandwidth utilization at the receivers. We use two sorts of topology. The first one is a dense-mode network where a large number of receivers is connected to a link. In such a scenario, each merging procedure must eliminate a large number of entries in each node. This sort of topology is intended to test the efficiency of the Direct Algorithm. The second type of network consists of a binary tree that attempts to emulate a sparse-mode topology. The idea is to observe how the Direct Algorithm responds to cascade scenarios and prove that the use of virtual layering is indicated in such networks.

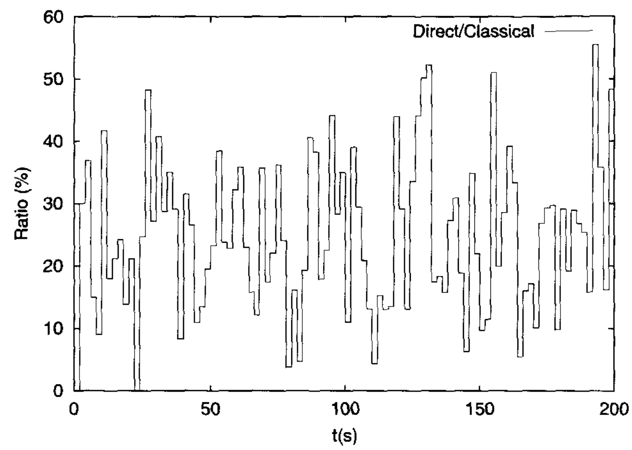
The metric we use to evaluate the efficiency of the algorithms is the global video degradation (or, inversely, the global video quality) at the receivers. For each receiver in the multicast session we compute the difference between the required video rate and the received video rate (for more details see Section II). We also evaluate the tradeoffs between the number of layers and the level of satisfaction of the multicast session.

For the dense-mode network, we use the topology illustrated in Fig. 5(a). It consists of one source and ten receivers connected through one node. In our simulations, all links have the same capacity $C = 1$ Mbps and identical delays $\tau = 10$ ms. Using the same delay for all links and symmetric networks makes our analysis independent of parameters other than the available bandwidth in the links. In each one of the links we introduce an exponential traffic source to simulate background traffic.

Fig. 6(a) shows the global video degradation for topology 1. Observe that the Direct Algorithm always results in the lowest



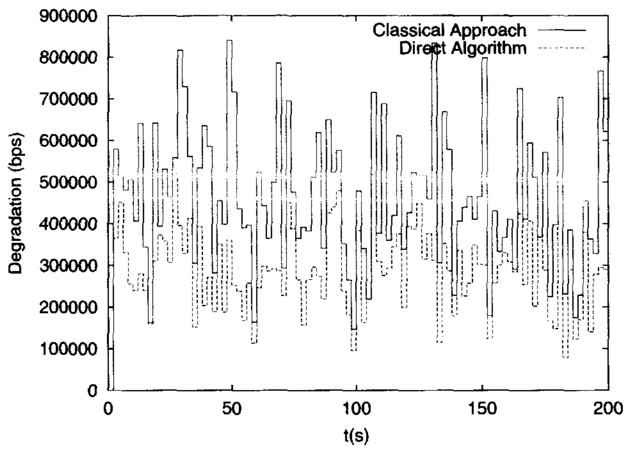
(a) Global video degradation. The results obtained by the Direct Algorithm are equivalent to the empirical border.



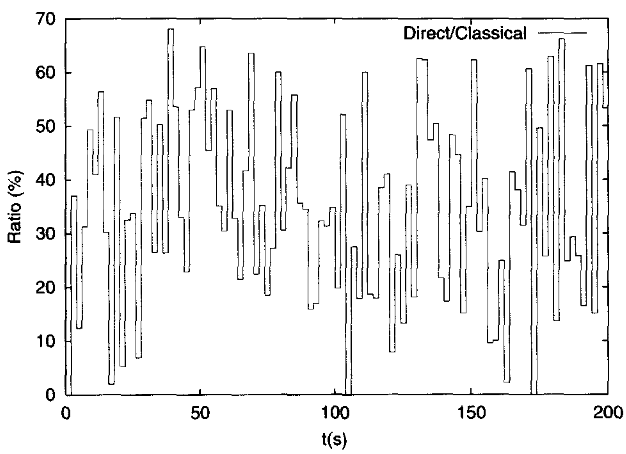
(b) Ratio between the Direct Algorithm and the classic approach.

Fig. 6. Results for topology 1.

global video degradation when compared with the classic approach. Fig. 6(b) depicts the ratio between the Direct Algorithm and the classic approach for the same topology. The average ratio is approximately 25%, which is quite large for a relatively small network with 10 receivers. Moreover, in such a scenario the Direct Algorithm results in the empirical border, i.e., the Direct Algorithm leads to the best possible global video quality. This is what we expected, because only one merging procedure is performed for each group of feedback control packets. In such networks where receivers are distributed in a dense way, the smaller the depth of the multicast tree, the closer is the Direct Algorithm to the empirical border. In any event, the Direct Algorithm is intrinsically more efficient than the classic approach in non-cooperative networks. We performed the same analysis for the sparse-mode topology with 8 receivers shown in Fig. 5(b). The simulation parameters are the same of the simulation of topology 1, i.e., symmetric links with capacity $C = 1$ Mbps and delay $\tau = 10$ ms. The global video degra-

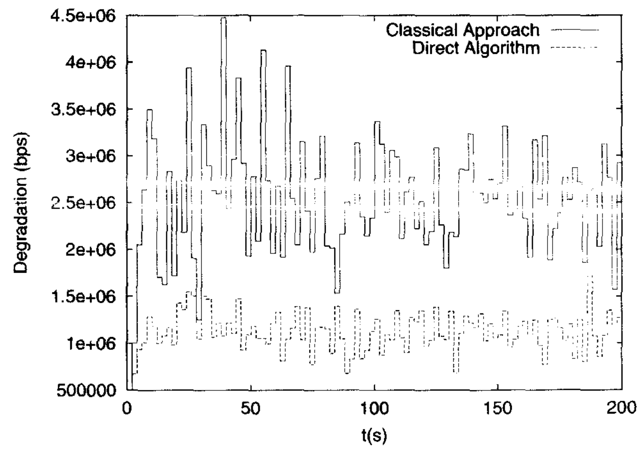


(a) Global video degradation.

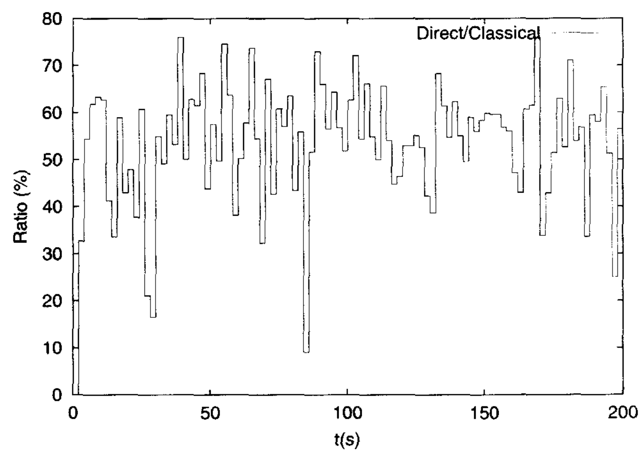


(b) Ratio between the Direct Algorithm and the classic approach.

Fig. 7. Results for topology 2.



(a) Global video degradation.

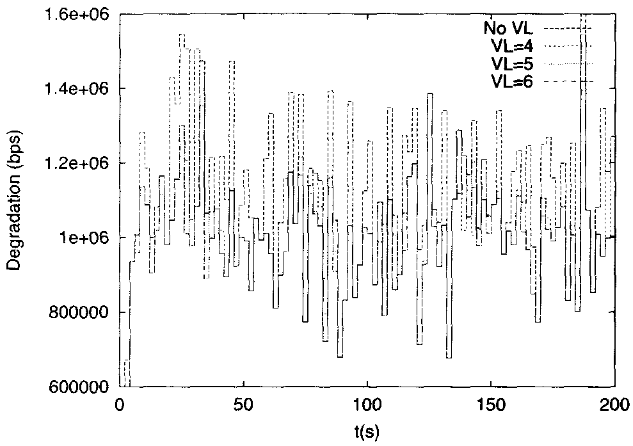


(b) Ratio between the Direct Algorithm and the classic approach.

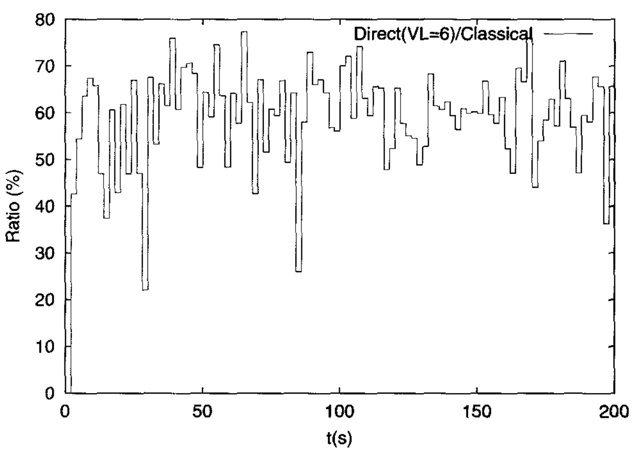
Fig. 8. Results for topology 3.

dation for the Direct Algorithm and for the classic approach are shown in Fig. 7(a). Note that the Direct Algorithm and the classic approach are closer than in the case of topology 1, but the classic approach is still bounded by the Direct Algorithm. The reasons why both approaches are closer are the number of discarded entries during the merging procedures at the nodes and the depth of the multicast tree. In the example of topology 1, when the node performs the merging procedure it must discard $c_T - L = 10 - 3 = 7$ entries. When executing this operation, the probability that the classic approach leads to erroneous entry discards is higher because it performs a larger number of consecutive steps. In the case of topology 2, the merging procedure is distributed in different nodes and at each one of them at the most $c_T - L = 6 - 3 = 3$ entries. As for the simulations of topology 1, Fig. 7(b) shows the ratio between the Direct Algorithm and the classic approach. Observe that even in this case the Direct Algorithm is about 35% more efficient than the classic approach in about 50% of the time. We now provide a deeper analysis of sparse-mode topologies. We simulate the network depicted in Fig. 5(c), which differs from topology 2 in the number of receivers (16 in the place of 8, which corresponds to one more level in the tree). The simulation parameters are the same of the previous simulations. The maximum number of entries discarded is also 3, as in the previous case, and the number of successive steps performed by the classic approach is at most 3. Nevertheless, the multicast tree has one extra level. It is likely that the advantages of the Direct Algorithm are even more pronounced. Figs. 8(a) and 8(b) show that this really happens and that with an increase of 1 in the depth of the multicast tree the Direct Algorithm leads to levels of global video degradation that are about 55% better than the ones of the classic approach.

We evaluate now the Direct Algorithm with virtual layers. We use in our simulation topology 3 of Fig. 5(c). We do not use the virtual layering technique in topology 1 because for this topology the Direct Algorithm always leads to the empirical border, so there is no advantage in introducing extra overhead. Since virtual layering tries to improve the quality by keeping extra information, the best results are expected to take place when the



(a) Global video degradation with virtual layering.



(b) Ratio between the Direct Algorithm with virtual layering and the classic approach.

Fig. 9. Results for topology 3 with virtual layering.

multicast tree is deeper. Fig. 9(a) shows the results for $\lambda = 4, 5,$ and 6. The number of video layers transmitted by the source is 3. As for the other simulations, we also show in Fig. 9(b) the ratio between the Direct Algorithm and the classic approach when we use 6 virtual layers. Note that a simple variable manipulation improves the global video quality of the multicast session.

V. CONCLUSIONS

In this paper, we have addressed the problem of optimization of feedback control information in order to improve the global video quality at the receivers in multicast layered communication. The use of feedback control packets allows the source to compute the video layers in such a way to provide all receivers with the fairest rates. This paper deals with a crucial problem of such systems: The feedback packet implosion when the multicast group is large. We proposed the Direct Algorithm that efficiently merge feedback control packets at intermediate nodes in order to optimize the global quality at the receivers. First, by performing the merging procedure in only one loop allows the node to be aware of all possible candidate packets to be forwarded to the upstream node. Contrary to the classic approach, where the merging procedure is performed in successive steps, the Direct Algorithm always leads to the best result in non-cooperative networks. Incidentally, to minimize the discarding of relevant information that intrinsically occurs in non-cooperative networks, the Direct Algorithm may use the concept of virtual layering.

We have analyzed and simulated the Direct Algorithm in two different types of network: Dense-mode, where an intermediate node can have a large number of upcoming links, and sparse-mode, where receivers are sparsely distributed throughout the network. The results show that the Direct Algorithm improves the global video degradation at the receivers in all cases. In sparse-mode topologies, the Direct Algorithm alway leads to better global video quality and tends to improve even more the results with the increasing of the depth of the multicast tree. Moreover, the results also show that the use of virtual layering allows the algorithm to reduce the influence of non-cooperation between sub-networks. In dense-mode networks, the advantages of using the Direct Algorithm are even more pronounced because the algorithm approximates the behavior of the empirical border.

REFERENCES

[1] B. J. Vickers, C. Albuquerque, and T. Suda, "Source-adaptive multi-layered multicast algorithms for real-time video distribution," *IEEE/ACM Trans. Networking*, vol. 8, pp. 720–733, Dec. 2000.

[2] X. Li, M. H. Ammar, and S. Paul, "Video multicast over the Internet," *IEEE Network Mag.*, Mar. 1999.

[3] M. D. de Amorim, O. C. M. B. Duarte, and G. Pujolle, "Multi-criteria arguments for improving the fairness of layered multicast applications," *Lecture Notes in Computer Science 1815*, (Paris, France), pp. 1–10, May 2000.

[4] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. Select. Areas Commun.*, vol. 15, no. 6, pp. 982–1001, Aug. 1997.

[5] D. Taubman and A. Zakhor, "Multi-rate 3-D subband coding of video," in *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 572–588, Sept. 1994.

[6] M. Vishwanath and P. Chou, "An efficient algorithm for hierarchical compression of video," in *Proc. IEEE Int. Conf. Image Processing*, (Austin, TX, USA), Nov. 1994.

[7] E. Amir, S. McCanne, and M. Vetterli, "A layered DCT coder for Internet video," in *IEEE Int. Conf. Image Processing*, (Lausanne, Suisse), Sept. 1996.

[8] J. Nonnenmacher and E. W. Biersack, "Scalable feedback for large groups," *IEEE Trans. Networking*, vol. 7, no. 3, pp. 375–386, June 1999.

[9] D. Saparilla and K. W. Ross, "Optimal streaming of layered video," in *Proc. INFOCOM'2000*, (Tel-Aviv, Israel), Mar. 2000.

[10] S. Sarkar and L. Tassiulas, "Distributed algorithms for computation of fair rates in multirate multicast trees," in *Proc. INFOCOM'2000*, (Tel-Aviv, Israel), Mar. 2000.

[11] D. Rubenstein, J. Kurose, and D. Towsley, "The impact of multicast layering on network fairness," in *Proc. SIGCOMM'99*, (Cambridge, Massachusetts, USA), Sept. 1999.

[12] R. Rejaie, M. Handley, and D. Estrin, "Layered quality adaptation for Internet video streaming," *IEEE J. Select. Areas Commun.*, vol. 18, no. 12, Dec. 2000.

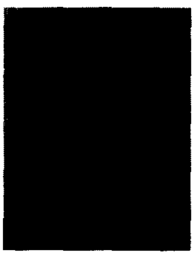
[13] A. Legout and E. Biersack, "PLM: Fast convergence for cumulative layered multicast transmission scheme," in *Proc. ACM SIGMETRICS'2000*, (Santa Clara, CA, USA), June 2000.

[14] B. J. Vickers, C. Albuquerque, and T. Suda, "Adaptive multicast of multi-layered video: Rate-based and credit-based approaches," in *Proc. INFOCOM'98*, (San Francisco, CA, USA), Mar. 1998.

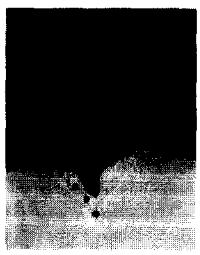
[15] R. Rejaie, *An End-to-End Architecture for Quality Adaptive Streaming Applications in the Internet*. PhD thesis, University of Southern California, Dec. 1999.

[16] J. C. Bolot and T. Turletti, "Experience with control mechanisms for packet video in the Internet," *ACM Computer Communication Review*, vol. 28, no. 1, pp. 4–15, Jan. 1998.
[17] S. Y. Cheung, M. H. Ammar, and X. Li, "On the use of destination set grouping to improve fairness in multicast video distribution," in *Proc. INFOCOM'96*, (San Francisco, CA, USA), Mar. 1996.
[18] S. McCanne, V. Jacobson, and M. Vitterli, "Receiver-driven layered multicast," in *Proc. SIGCOMM'96*, (Stanford, CA, USA), Aug. 1996.
[19] L. H. M. K. Costa, S. Fdida, and O. C. M. B. Duarte, "Hop-by-hop multicast routing protocol," in *Proc. SIGCOMM'2001*, (San Diego, CA, USA), Aug. 2001.
[20] J. Moy, "Multicast extensions to OSPF," RFC 1584, May 1994.
[21] K. Fall and K. Varadhan, *The ns Manual*. UC Berkeley, LBL, USC/ISI, and Xerox Parc, Jan. 2001. Available at http://www.isi.edu/nsnam/ns/ns-documentation.html.

Otto Carlos M. B. Duarte was born in Rio de Janeiro, Brazil, on October 23, 1953. He received the Electronic Engineer degree and the M.Sc. degree in Electrical Engineering from the Federal University of Rio de Janeiro, Brazil, in 1976 and 1981, respectively, and the Dr.Ing. degree from ENST/Paris, France, in 1985. Since 1978, he is Associate Professor at the Federal University of Rio de Janeiro. From January 1992 to June 1993, he has worked at MASI Laboratory in the University of Paris 6. In 1995, he has spent three months at International Computer Science Institute (ICSI) associated to the University of California at Berkeley. Presently, he is heading the computer network group (Grupo de Teleinformática e Automação - GTA). His major research interests are in high speed communications, mobility, security and QoS guarantees.



Marcelo Dias de Amorim received the "Cum Laude" Degree in Electronic Engineering from the Federal University of Rio de Janeiro (UFRJ), Brazil, in 1996, and the M.Sc. Degree in Electrical Engineering from COPPE/UFRJ, Brazil, in 1998. He received his Ph.D. degree with honors in Computer Science from the University of Versailles, France, in 2001. He spent one year at LRI Laboratory of the University of Paris Sud. He is currently a research scientist in the Network and Performance Group of the LIP6 Laboratory (CNRS - University of Paris 6) and in EuronetLab, the research laboratory dedicated to next-generation Internet, France. His major research interests are in QoS guarantees, mobility, peer-to-peer networks, and large-scale routing.



Guy Pujolle received his Ph.D. and Thèse d'Etat degrees in computer science from the University of Paris IX and XI in 1975 and 1978, respectively. He is currently professor at the University of Paris VI. He is chair of IFIP Working Group 6.2 on Network and Internetwork Architectures and is a governor of ICCC. He has published widely in the areas of computer systems modeling and performance, queuing theory, and high-speed networks. His research interests include the analysis and modeling of data communication systems, protocols, high-performance networking, intelligence in networking, and wireless networks. He has been Professor Honoris Causa of Beijing University since 1988. He was awarded the Special Seymour Cray Award in 1991 for his research and the Silver Core from IFIP in 1995.