Virtual Layering and Efficient Merging in Non-Cooperative Multicast Trees

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

[1]Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie
8, rue du Capitaine Scott
75015 – Paris - France
{Marcelo.Amorim,Guy.Pujolle}@lip6.fr

[2]Grupo de Teleinformática e Automação
COPPE – Univ. Federal do Rio de Janeiro
Cx. P. 68504 – 21945-970
Rio de Janeiro – RJ Brasil
otto@gta.ufrj.br

Abstract *A critical problem faced by feedback-merger mechanisms is the lack of information that is discarded due to the hidden nodes in multicast trees. A node is said to be hidden from another if it is located in a sub tree that is a result from a fork in any upstream node. We propose in this paper the virtual layering scheme to avoid the problem caused by hidden-nodes in multi-layered multicast video environments. The virtual layering scheme induces intermediate nodes to keep extra states of the multicast session, which reduces the video degradation for the whole set of receivers. Furthermore, this scheme is coupled with the Direct Algorithm in order to improve the degree of satisfaction at heterogeneous receivers. The algorithm relies on a mechanism that dynamically controls the rates of the video layers and addresses scalability issues by implementing a merging procedure at intermediate nodes in order to avoid packet implosion at the source. The Virtual Layering scheme combined with the Direct Algorithm is optimized to achieve high global video quality and reduced bandwidth requirements. The results show that the proposed scheme leads to improved global video quality at heterogeneous receivers with no cost of extra bandwidth.*

Keywords: *Virtual layering, adaptive applications, multi-layering, multicast, video quality*

1 Introduction

Heterogeneity in intermediate and terminal equipments is a major problem for the deployment of multicast applications in current network architectures such as the Internet. In a multicast session, if the source sends at a rate compatible with the slowest receivers, the paths to the faster receivers will be underutilized. On the other hand, if the source sends at high rates, slower clients will not be able to receive data. These challenges are intrinsic to video distribution, since most video applications are multicast by nature [1,2,3].

A number of adaptive systems for managing the source rates in multicast networks have been proposed [1,2,4,5,6,7]. They do provide QoS improvements but do not (or almost not) explicitly deal with the problem of computing the exact rate of the source in order to optimize the video quality for the whole set of receivers. 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 [8,9,10,11,12]. 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. Nevertheless, this approach is followed by scalability issues if the number of receiver is large [4,5,13,14]. When all packets sent by the receivers arrive at the source, a feedback implosion occurs and the system suffers from a source collapse.

Our system uses the concept of feedback mergers and implements a procedure at intermediate nodes that combines concurrent feedback control packets in order to avoid feedback implosion at the source. Nevertheless, feedback mergers have an intrinsic drawback. When performing a merging procedure, a node does not dispose of information that has been discarded in two or more downstream nodes. To avoid this kind of problem, we propose 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 confrontation with control information of neighbor sub networks. This means that the virtual layering scheme reduces the probability of eliminating important control information during a merging procedure. Our analyses show that this approach leads to results that are close to the empirical border. The empirical border is obtained when no merging procedures are performed at intermediate nodes. In this case, all information sent by the receivers arrives at the source. Thus, the source can compute the video layers in such a way as to obtain the highest possible global video quality.

We also propose in this paper the Direct Algorithm, implemented in such a way 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.

The remainder of this paper is organized as follows. In Section 2 we present the system architecture and the objectives of the paper. Section 3 describes the negative effects caused by hidden-nodes in multicast systems with feedback mergers. In Section 4 we propose and analyze the Direct Algorithm. Section 5 presents the Direct Algorithm with virtual layering. Several analyses and simulations to evaluate the proposed schemes are described in Section 6 and Section 7 presents some related works. Finally, Section 8 concludes this paper.

2 System architecture

We use in this paper a network architecture consisting of one source and a number of receivers belonging to a multicast session. The source distributes video to the receivers using the multi-layered approach. In multi-layered video schemes, the raw video is encoded in one or more layers, say a base layer and one or more enhancement layers. The global quality of the retrieved video depends on the number of enhancement layers that are incrementally combined to the base layer. The layers follow a causal law, i.e., a layer can compose a video only if all of the lowest priority layers in the hierarchy are available. The use of multi-layered video in heterogeneous multicast networks is a good solution because receivers with different receiving capacities join only the layers they are able to receive. Figure 1 shows a simple multicast session where the source is able to send three video layers. As we can see, the multicast session is fair in the sense all the receivers are provided with the rates they are able to receive.

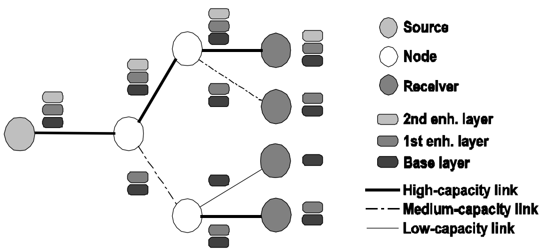


Figure 1 - A multicast session using multi-layered video

As the number of receivers in the multicast session increases, the source must also increase the number of video layers in order to supply the receivers with the required rate. Nevertheless, the number of video layers transmitted by the source cannot be unlimited.

Let $\mathbf{R}=\{r_1, r_2, \ldots, r_N\}$ be the set of N receivers in the multicast session. Receiver r_i has an associated receiving capacity b_{ri}, given by the minimum available bandwidth in the path from the source to r_i:

$$\square\,{}^{T_j}_{x_i} \square \frac{S^{T_j}_{x_i}}{1 \square S^{D}_{x_i} \frac{\square x_i}{x_i}} \tag{1}$$

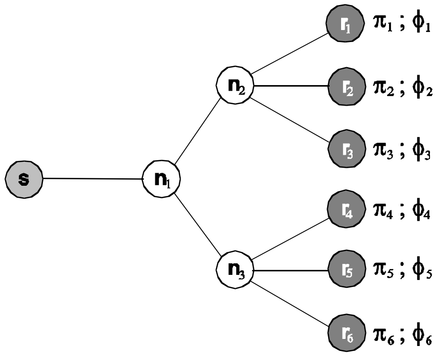
where $L_{s,ri}$ is the set of links traversed by the control packet from the source s to the receiver r_i and $c(l)$ is the available bandwidth in link $l \in L_{s,ri}$.

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 soon as an inspection packet arrives at the receiver, it builds a return feedback packet containing its receiving capacity and sends it back to the source. When routing 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, we should use robust multicast routing algorithms like HBH [15] or MOSPF [16].

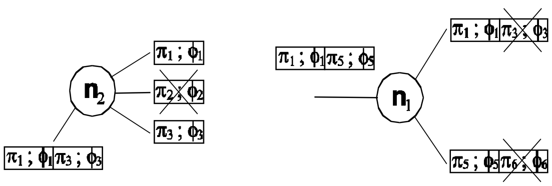
A feedback control packet has k entries, $1 \leq k \leq L$, where L is the number of video layers transmitted by the source. Each entry e_i, $i=1,\ldots,k$, has two fields: a video rate π_{ei}, and the number of receivers φ_{ei} that require the rate π_{ei}. Our system is similar to the control scheme proposed in [1,17] (we will call this scheme the "classical approach"), where the arriving feedback packets are stored in a temporary array $\boldsymbol{t'}$ 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 $\pi_{e(i-1)} \leq \pi_{ei} \leq \pi_{e(i+1)}$, $\forall i$. If $\pi_{ei}=\pi_{e(i+1)}$, then e_i and e_{i+1} are automatically combined by performing $\pi_{ei}^{+} = \pi_{ei}^{-}$ and $\varphi_{ei}^{+} = \varphi_{ei}^{-} + \varphi_{e(i+1)}^{-}$, where the symbols - and + represent the values of the variables respectively before and after the merging procedure. In practice, two entries are automatically merged if the difference $\pi_{e(i+1)}-\pi_{ei}$ is within a certain threshold α. In this case, we say that these entries are "compatible". Further details in the merging procedures are presented in Section 4.

3 The fork problem

One of the problems intrinsic to feedback-based schemes in multicast trees is the *hidden-neighborhood*. This problem is particularly crucial when the network uses feedback mergers at intermediate nodes to reduce the probability of implosion at the source, subject of this paper. We illustrate this situation with a simple example. Consider the multicast session shown in Figure 2(a). We enumerate intermediate nodes from n_1 to n_3 and receivers from r_1 to r_6. Suppose that a feedback packet must have at most two entries. Thus, since nodes n_2 and n_3 have three incoming feedback packets (with one entry each), they must discard one packet (see Figure 2(b)). Consequently, node n_1 must discard two entries because it receives two packets with two entries each.



(a) Example of multicast session



(b) Entry discard

Figure 2 Merging procedure

The information discarded by node n_2 is never confronted with the information discarded by node n_3. This is because n_2 and n_3 belong to different sub networks (due to the forking at node n_1). We will see now how this lack of integration among sub networks can affect the efficiency of the merging procedure.

Suppose our merging algorithm performs as follows:

- The total number of entries is T_e is given by the sum of the number of entries of every incoming packet.
- The intermediate node discards T_e-L entries, where L is the maximum number of entries accepted by the source. Thus, every feedback packet that arrives at the source has at most L entries.
- The algorithm is performed in a loop where the node discards one entry at a time. The node discards the entry e_i that leads to the smallest *hosting-loss* Ω, where $\Omega=\phi_i(\pi_{ei}-\pi_{e(i-1)})$, for $i=2,...,T_e$.

In the scenario of Figure node n_2 has discarded entry e_2 and node n_3 has discarded entry e_4. When the resulting feedback packets arrive at node n_1, this information is no longer available.

Suppose now that nodes n_2 and n_3 do not perform the merging procedure. Node n_2 will then have two incoming packets with three entries each. The number of discarded entries will be four. Consider also that $\pi_{e2}=\pi_{e4}$. In this case, the algorithm automatically combines these entries. The new entry (result of this combination) will have $\pi_{e(new)}=\pi_{e2}=\pi_{e4}$ and $\varphi_{e(new)}=\varphi_{e2}+\varphi_{e4}$. With these new quantities, it is possible that the hosting-loss will be greater than the hosting-loss of the other entries. As a consequence, the final feedback packet sent to the source will be different from the result shown in Figure 2, because this entry will not be discarded. This problem is subject of Section 5. We first present in next section a detailed description of the merging algorithm.

4 The merging algorithm

When performing the merging procedure, a node must follow some rules in order to discard the entries that lead to an improved global video quality at the receivers. The global video quality can be defined in many ways [18]. Here, we use the following rules:

Rule 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.*

Rule 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.*

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

$$\square_{x_i}^{T_j} \square \frac{S_{x_i}^{T_j}}{1 \square S_{x_i}^{D} \frac{\square x_i}{x_i}} \qquad \textbf{(2)}$$

where N is the total number of receivers in the multicast session.

4.1 Classical approach

The classical 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 $G = \Sigma^{c_T}_{i=1,\, i \neq j} \pi_{ei}\, \varphi_{ei}$, where c_T is the size of the temporary array.

The number of destinations φ_{ej} 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.

4.2 Limitations of the classical approach

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 scenario previously shown in Figure 2(a) with the source transmitting two video layers (L=2). Consider also G_{ej} to be the *goodput* rate when discarding entry e_j. At the node n_1, 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.

Let **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

$$\square_{x_i}^{T_j} \square \frac{S_{x_i}^{T_j}}{1 \square S_{x_i}^{D} \frac{\square x_i}{x_i}} \qquad \textbf{(3)}$$

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

$$\square_{x_i}^{T_j} \square \frac{S_{x_i}^{T_j}}{1 \square S_{x_i}^{D} \frac{\square x_i}{x_i}} \qquad \textbf{(4)}$$

where $\mathbf{O}_x$ is chosen in such a way that $\mathbf{O}_x \leq \pi_k < \mathbf{O}_{x=1}$. Nevertheless, the classical 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

$$\square_{x_i}^{T_j} \square \frac{S_{x_i}^{T_j}}{1 \square S_{x_i}^{D} \frac{\square x_i}{x_i}} \qquad \textbf{(5)}$$

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

$$\square_{x_i}^{T_j} \square \frac{S_{x_i}^{T_j}}{1 \square S_{x_i}^{D} \frac{\square x_i}{x_i}} \qquad \textbf{(6)}$$

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

$$\square_{x_i}^{T_j} \square \frac{S_{x_i}^{T_j}}{1 \square S_{x_i}^{D} \frac{\square x_i}{x_i}} \qquad \textbf{(7)}$$

By combining equations (4) and (7), and after some manipulation, we have that the classical approach will not lead to the fairest output packet if

$$\square_{x_i}^{T_j} \square \frac{S_{x_i}^{T_j}}{1 \square S_{x_i}^{D} \frac{\square x_i}{x_i}} \qquad \textbf{(8)}$$

4.3 Direct Algorithm

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 ($l_1,\ldots,l_L$) and at some intermediate node the packet that must be reduced has c_T entries, the Direct Algorithm works as follows. We make $l_1 = \pi_{e1}$ and we compute $l_2,\ldots,l_L$ so as to minimize the *loss sum*

$$\square_{x_i}^{T_j} \square \frac{S_{x_i}^{T_j}}{1 \square S_{x_i}^{D} \frac{\square x_i}{x_i}} \qquad \textbf{(9)}$$

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)
Begin
 δ = 0;
 δtmp = 0;
 Otmp = "first candidate packet";
 While (∃ "next candidate packet")
 Begin
  For (k=1;k≤cT;k++)
   δtmp = δtmp + φk(πk-πOx);
  If (δtmp<δ) then
  Begin
   δ = δtmp;
   O = Otmp;
  End
```

```
  otmp = "next candidate packet";
 End
End
```

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 $p^n{}_i$, $i=1,... k$, 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 we make the following assumptions:

- All incoming links $p^n{}_i$, $i=1,...,k$, 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.

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

$$\square^{T_j}_{x_i} \square \frac{S^{T_j}_{x_i}}{1 \square S^{D}_{x_i} \frac{\square x_i}{x_i}} \quad \textbf{(10)}$$

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

$$\square^{T_j}_{x_i} \square \frac{S^{T_j}_{x_i}}{1 \square S^{D}_{x_i} \frac{\square x_i}{x_i}} \quad \textbf{(11)}$$

Figure 3 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 Figure 3, 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 a hybrid approach performed in two steps. During the first one, the merging procedure uses the classical 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 keeps the complexity of the system under control and allows the merging procedure to improve the quality of the multicast session. Figure 4 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 Figure 3, we note that the number of candidate packets can be easily controlled by applying the hybrid approach.

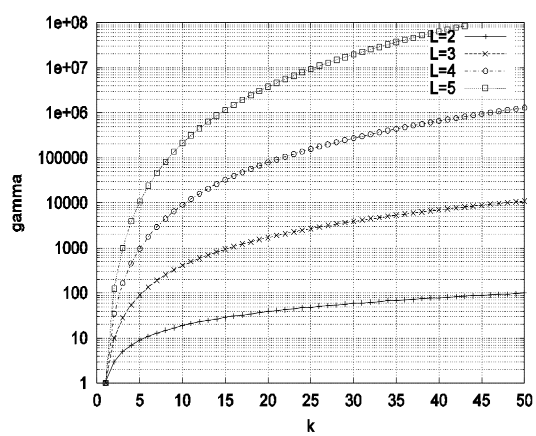


Figure 3 Responsiveness of γ to k and L

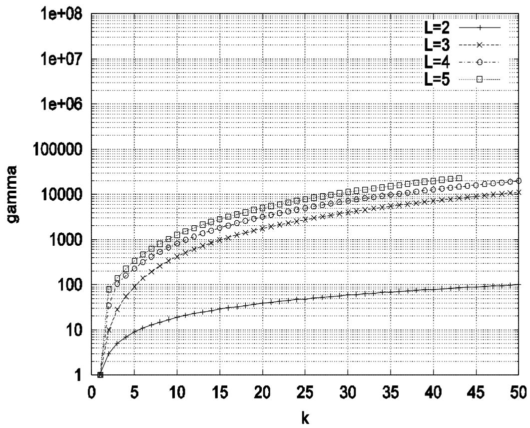


Figure 4 Responsiveness of γ to k and L for the hybrid approach

5 Virtual layering

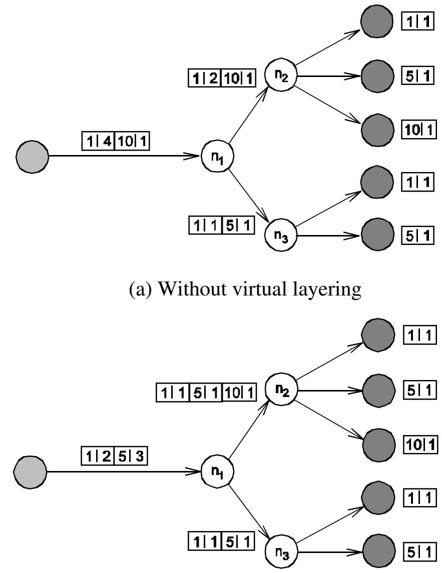
Depending on the number of intermediate nodes that perform the merging algorithm, entries that improve the global video quality may be inappropriately discarded. When performing a merging procedure, intermediate nodes only handle information sent by the receivers in their respective downstream sub trees. Receivers in other sub trees have no influence on the candidate output packets, and this is the price to reduce signalling 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 in what manner the virtual layering scheme can reduce the limitations of non-cooperative sub trees.

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

$$\square_{x_i}^{T_j} \square \frac{S_{x_i}^{T_j}}{1 \square S_{x_i}^{D} \frac{\square x_i}{x_i}} \qquad \textbf{(12)}$$

δ_{n0} is the global video degradation in node n_0 when the fairest output feedback packet for node n_j, $j=1,...,g$, is $\mathbf{O}_{nj}$. It is possible that $\exists\ \mathbf{O}'_{nj}$, $j=1,...,g$, $\mathbf{O}_{nj} \neq \delta'_{nj}$, such that after applying the virtual layering scheme we have $\delta'_{n0} < \delta_{n0}$.



(a) Without virtual layering

(b) Using one virtual layer

Figure 5 Evolution of the merging procedure

Consider the situation depicted in Figure 5, 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 sub tree that begins at node n_3 also contains an entry with the same rate $\pi=5$. If the same entry were not eliminated at node n_2, the algorithm at node n_1 would not discard $\pi=5$ but $\pi=10$.

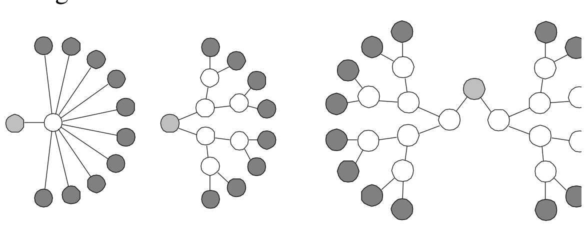
The virtual layering scheme reduces the probability that this sort of problem happen. In the same example above, if we do $\lambda=3$, the merging algorithm is not performed at node n_2. Consequently, the entry that corresponds to the receivers that want to receive $\pi=5$ is not discarded. At node n_1,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

6 Analysis

In this section, we analyze the virtual layering scheme and the Direct Algorithm and compare its performance with the classical approach using the NS-2 network simulator [19]. We evaluate the level of video degradation and the network bandwidth utilization at the receivers. We use two types 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 type 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 the behavior of the Direct Algorithm in face of 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. 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 Figure 6(a). It consists of one source and ten receivers connected through one node. In our simulations, all links have the same capacity C=1Mbps 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.



(a) (b) (c)

Figure 6 Simulation topologies (a) Dense mode; (b) Binary – 8; (c) Binary - 16

Figure 7 depicts the ratio between the Direct Algorithm and the classical approach for topology 1. Observe that the Direct Algorithm always results in the lowest global video degradation when compared with the classical approach.. The average ratio is ≈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 classical approach in non-cooperative networks.

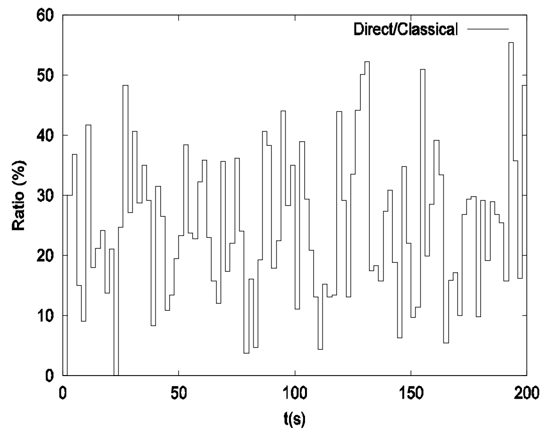


Figure 7 Ration between the Direct Algorithm and the classical approach for topology 1

We performed the same analysis for the sparse-mode topology with 8 receivers shown in Figure 6(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 ratio between the Direct Algorithm and the classical approach is shown in Figure 8.

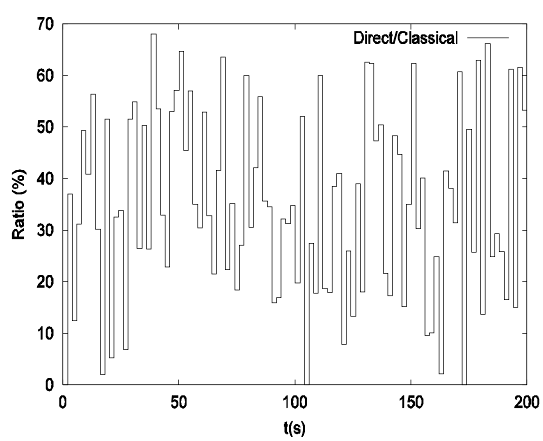


Figure 8 Ratio between the Direct Algorithm and the classical approach for topology 2

Note that the Direct Algorithm and the classical approach are closer than in the case of topology 1, but the classical 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 classical approach lead 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. Observe that even in this case the Direct Algorithm is about 35% more efficient than the classical approach in about 50% of the time.

We now provide a deeper analysis of sparse-mode topologies. We simulate the network depicted in Figure 6(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 classical 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. Figure 9 shows 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 classical approach.

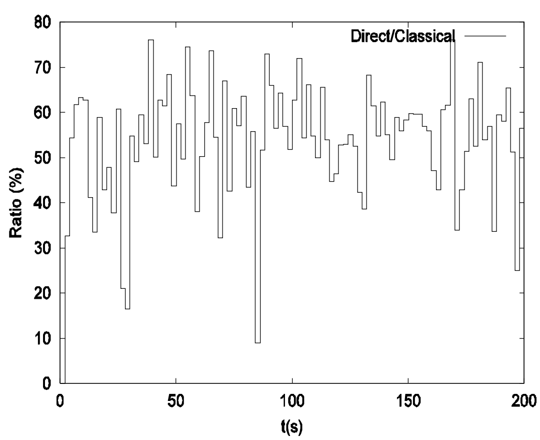


Figure 9 Ratio between the Direct Algorithm and the classical approach for topology 3

We evaluate now the Direct Algorithm with virtual layers. We use in our simulation topology 3 of Figure 6(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 multicast tree is deeper. Figure 10 shows the ratio between the Direct Algorithm and the classical approach when we use 6 virtual layers. Note that a simple variable manipulation improves the global video quality of the multicast session.

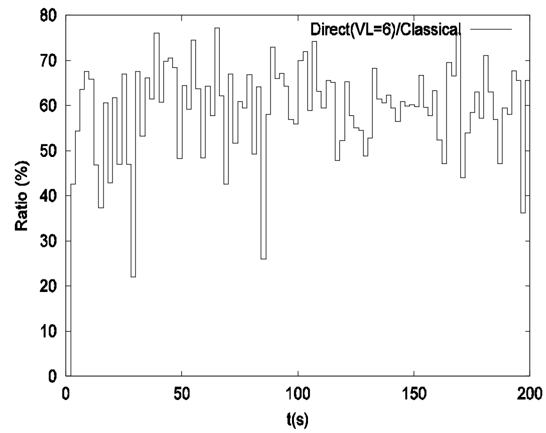


Figure 10 Ratio between the Direct Algorithm and the classical approach for topology 3 with virtual layering

7 Related works

Previous works have proposed some adaptive schemes to deal with the real-time aspect of video distribution [1,6,7,17,20,21,22,23]. In [1,6], 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 [22], 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 [23], 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 [6], the source generates a certain number of video layers and transmits each layer to a different IP-multicast group. The receivers subscribe to the number of groups they will. The communication is dynamic in the way receivers can dynamically join or leave groups. Nevertheless, they are limited to the layers the source decides to transmit.

8 Conclusions

In this paper, we have addressed two crucial problems for multicast layered communications based on feedback mergers: the non-cooperation among receivers in hidden sub networks and the 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. Nevertheless, in order to avoid feedback implosion at the source, feedback packets must be combined at intermediate nodes. We proposed the Direct Algorithm that efficiently merges 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 classical approach, where the merging procedure is performed in successive steps, the Direct Algorithm and the virtual layering scheme always improve the results in non-cooperative networks.

We have analyzed and simulated the virtual layering scheme and 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 our proposal improves the global video degradation at the receivers in all cases. In sparse-mode topologies, our scheme always leads to a 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 are even more pronounced because the algorithm approximates the behavior of the optimal solution that is the empirical border.

Acknowledgements

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

References

[1] B. J. Vickers, C. Albuquerque, and T. Suda, Source-adaptive multi-layered multicast algorithms for real-time video distribution, *IEEE/ACM Transactions on Networking*, 8(6):720-733, December 2000.

[2] X. Li, M. H. Ammar, and S. Paul, Video multicast over the internet, *IEEE Network Magazine*, March 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, in *Networking 2000*, (Paris, France), May 2000.

[4] S. Sarkar and L. Tassiulas, Distributed algorithms for computation of fair rates in multirate multicast trees, in *IEEE Infocom*, (Tel-Aviv, Israel), March 2000.

[5] D. Rubenstein, J. Kurose, and D. Towsley, The impact of multicast layering on network fairness, in *ACM Sigcomm*, (Cambridge, Massachusets, USA), September 1999.

[6] S. McCanne, V. Jacobson, and M. Vitterli, Receiver-driven layered multicast, in *ACM Sigcomm*, (Stanford, CA, USA), August 1996.

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

[8] S. McCanne, M. Vetterli, and V. Jacobson, Low-complexity video coding for receiver-driven layered multicast, *IEEE Journal on Selected Areas in Communications*, 15(6):982-1001, Aug. 1997.

[9] D. Taubman and A. Zakhor, Multi-rate 3-D subband coding of video, *IEEE Transactions on Image Processing*, 3(5):572-588, September 1994.

[10] M. Vishwanath and P. Chou, An efficient algorithm for hierarchical compression of video, in *IEEE International Conference on Image Processing*, (Austin, TX, USA), November 1994.

[11] E. Amir, S. McCanne, and M. Vetterli, A layered DCT coder for Internet video, in *IEEE International Conference on Image Processing*, (Lausanne, Suisse), September 1996.

[12] P. A. S. Gonçalves, J. F. de Rezende, O. C. M. B. Duarte, and G. Pujolle, Optimal feedback for quality source-adaptive schemes in multicast multi-layered video environments, in *IFIP Networking*, (Pisa, Italy), May 2002.

[13] J. Nonnenmacher and E. W. Biersack, Scalable feedback for large groups, *IEEE Transactions on Networking*, 7(3):375-386, June 1999.

[14] D. Saparilla and K. W. Ross, Optimal streaming of layered video, in *IEEE Infocom*, (Tel-Aviv, Israel), March 2000.

[15] L. H. M. K. Costa, S. Fdida, and O. C. M. B. Duarte, Hop-by-hop multicast routing protocol, in *ACM Sigcomm*, (San Diego, CA, USA), August 2001.

[16] J. Moy, Multicast Extensions to OSPF. RFC 1584, May 1994.

[17] 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), March 1998.

[18] M. D. de Amorim, O. C. M. B. Duarte, and G. Pujolle, Application-aware multicast, in *IEEE Globecom*, (San Antonio, TX, USA), November 2001.

[19] K. Fall and K. Varadhan, The ns Manual. UC Berkeley, LBL, USC/ISI, and Xerox Parc, January 2001. http://www.isi.edu/ nsnam/ns/ns-documentation.html.

[20] R. Rejaie, M. Handley, and D. Estrin, Layered quality adaptation for internet video streaming, *IEEE Journal on Selected Areas in Communications*, 18(12), December 2000.

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

[22] J. C. Bolot and T. Turletti, Experience with control mechanisms for packet video in the internet, *ACM Computer Communication Review*, 28(1):4-15, January 1998.

[23] 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), March 1996.