QoS mechanisms for multilayered multicast applications

Paulo André da S. Gonçalves, Marcelo D. de Amorim,
José F. de Rezende, and Otto Carlos M. B. Duarte

Grupo de Teleinformática e Automação – GTA
Programa de Engenharia Elétrica – COPPE
Universidade Federal do Rio de Janeiro – UFRJ
P.O. Box 68504 – 21945-970 – Rio de Janeiro – Brasil
{apaulo,amorim,rezende,otto}@gta.ufrj.br

Abstract

New multilayered multicast applications have been emerged to deal with Quality of Service (QoS) requirements of heterogeneous receivers. Such applications introduce different mechanisms to accomplish their purposes such as the employment of implosion avoidance and quality control schemes, and the adoption of new feedback semantics. In this paper, we address the development of mechanisms that yield enhancements in quality/fairness of multilayered multicast sessions. For this purpose, we propose a quality control scheme that allows for the density of satisfied users, the weighted bandwidth allocation, and the perceived quality at receivers. The feedback implosion problem is analyzed considering a feedback aggregation approach and a cooperative scheme among receivers. Our analysis shows that feedback aggregation is a potential solution to avoid feedback implosion while allowing the use of novel feedback semantics.

Keywords: Multicast, multilayered source, feedback suppression, application quality.

1 Introduction

Providing quality of service (QoS) to multicast communications faces a number of difficulties in the network infrastructure as well as in the end-system deployment. Such systems have been studied for many years, but a number of issues remain open. In this paper, we address the development of QoS mechanisms for multicast applications.

One of the main drawbacks encountered by multicast applications in the context of large scale applications is the heterogeneity of intermediate and terminal equipments. Indeed, multicast data may traverse high or low speed lines, congested nodes, wired or wireless links, and so on. Similarly, terminals can also be completely different, in terms of software or hardware capacities. Current multicast solutions, like IP-Multicast, do not take all these parameters into account.

These challenges are intrinsic to many multimedia applications because of their multicast nature. To deal with these problems, several strategies have been proposed to improve and guarantee a minimum quality of service. One approach is to use resource reservation in order to provide a certain level of guarantee [1]. An efficient resource reservation, however, requires a previous knowledge of the traffic characteristics, which is not always feasible. Another technique is to conceive adaptive multimedia applications. Introducing adaptability is seen as a key feature for the next generation of multimedia applications. This approach consists in giving to the application a certain level of consciousness on the network condition [2, 3, 4, 5, 6, 7]. In [3], 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 observe the available bandwidth and computes the minimum signal compression rate that should be applied in order to maximize the bandwidth utilization. In the hierarchical coding, each signal is coded in a base layer and in one or more improvement layers. These layers are correlated according to a causal law, and the quality of the presentation depends on the number of layers that are retrieved. The utilization of multilayered sources, however, introduces several challenges in the deployment of a multicast communication system, as for example in the context of multicast routing [8].

In this paper, we propose a number of mechanisms to address QoS issues in multilayered multicast communications. These mechanisms can be divided in three main groups. In the first part, we consider some techniques to improve the quality of the multicast communication from the user's point of view. We propose, describe, and analyze some approaches mainly for computing the rates of the layers. The second part is reserved for the feedback implosion problem. Since our mechanisms are based on control information exchanged between the source and the receivers, feedback implosion becomes a crucial problem. We develop a technique which improves the level of packet filtering at intermediate nodes. This is fundamental for an efficient use of the QoS mechanisms proposed in the context of this work. Finally, we consider the general case of passive networks. In a passive network, we are unable (or it is difficult) to implement new mechanisms in intermediate nodes, such as in the case of the Internet. We propose then a mechanism for feedback suppression at the receivers. We show through a number of analysis and simulations that our proposals lead to good improvements.

This paper is organized as follows. In Section 2, we present the context of this work. In Section 4, we present and analyze a number of quality control schemes for multilayered multicast communication. We address the feedback implosion problem and propose a number of solutions for improving feedback merging in Section 3. Also in Section 3, we propose a cooperative protocol for feedback suppression at the receivers. Finally, in Section 6, we conclude this paper.

2 Adaptive multicast applications

Adaptive applications seem to be well suited for the heterogeneous context of the Internet. The main idea is to adapt the quality according to the capacities of the receivers and the available capacities on the paths from sources to receivers. In this paper we focus on multicast continuous-media applications such as video, audio, or voice. These media have tight time constraints, accept some loss, and can be compressed. Furthermore, we consider the Internet multicast addressing scheme, where a message sent to an IP-Multicast address reaches all session members.

We can classify multicast adaptive applications in three main groups: adaptive single-flow, multicoding, and multilayering.

2.1 Adaptive single-flow

In the adaptive single-flow approach, the source transmits one flow to a single IP-multicast address. Thus, if the source sets its rate to the slowest receiver, fast receivers will experience a misuse of network resources. On the other hand, if the source sends at a higher rate, slow receivers will be unable to receive data. As a consequence, if we prioritize the quantity of receivers, the quality of the application is adjusted by the worst case. The adaptation is accomplished by the source, which adjusts the sending rate based on reports sent by the receivers. The source estimates the average quality of the communication and the congestion level of the network to adjust the sending rate. Fig. 1 illustrates the adaptive single source scheme.

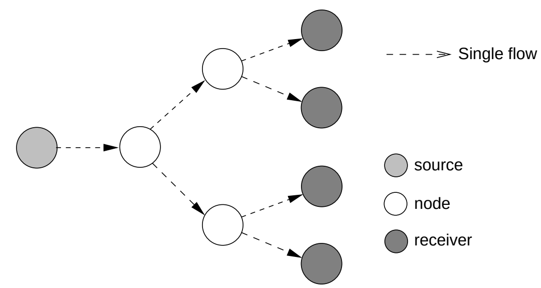


Figure 1: A single flow is sent to all the receivers no matter the available capacities of the receivers.

One of the first proposals using the concept of the adaptive single source is IVS (INRIA Video-conference System), proposed by Bolot, Turletti, and Wakeman [9, 10]. The rate control mechanism in IVS is implemented inside a H.261 coder. For adjusting the source rate, three parameters are considered: refresh rate, quantifier, and movement detection threshold. The specific applications need to determine which parameters must be modified in order to adjust the output source rate. Feedback information are obtained from the loss rate observed by the receivers, which uses RTP (Real Time Protocol) sequence numbers [11].

In order to avoid feedback implosion at the source, the authors propose a scalable request mechanism. Every receiver sends to the source a control packet if some conditions are respected. These conditions are a probability p and a frequency f which depend on the number of members in the multicast session.

2.2 Multicoding

In the multicoding approach, also called simulcast [12], the source maintains a certain number of flows derived from the same raw data. The receivers are classified according to their capabilities and each flow is addressed to a particular group. Over the Internet, these flows are sent to different IP-multicast addresses. When a receiver wishes to receive a flow, it joins the multicast group which transmits at a rate as close as possible to its receiving capacity. Fig. 2 shows the multicoding scheme.

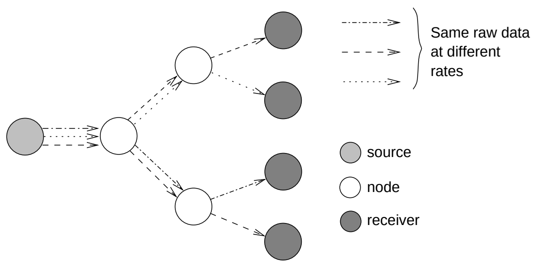


Figure 2: The same raw data are coded at different rates and sent to different multicast addresses.

The DSG (Destination Set Grouping) algorithm [12] is an example of multicoding scheme. This algorithm combines sender-driven and receiver-driven control mechanisms in order to improve the overall fairness of the multicast session and to obtain scalability. Fairness is obtained by transmitting multicoded data and the scalability is reached because the decision of subscription to multicast groups is made only by the receivers. The receivers are constantly informed about the network state. Thus, the receivers can make the decision to change or not of multicast group. DSG has two main components: (a) an intra-flow protocol used by the receivers belonging to the same multicast group in order to adjust the rate of transmission according to some predefined threshold and (b) an inter-flow protocol used by the receivers to change from multicast groups.

Whereas this approach lead to good fairness, it may be inefficient in terms of bandwidth utilization due to its redundancy of information.

2.3 Multilayering

The hierarchical multilayered approach tries to solve the heterogeneity problem and, at same time, avoid the waste of bandwidth. In the hierarchical coding, each signal is coded in a base layer and in one or more improvement layers. Each layer is sent out on a different IP-multicast address. These layers are correlated according to a causal law, and the quality of the presentation depends on the number of layers that are retrieved by the receivers. Thus, receivers with different receiving capacities can choose only the layers they are able to receive. In this scenario, the nodes join a number of multicast groups in order to adapt their perceived quality to their capacities. Fig. 3 shows a multilayer multicast session with heterogeneous receivers.

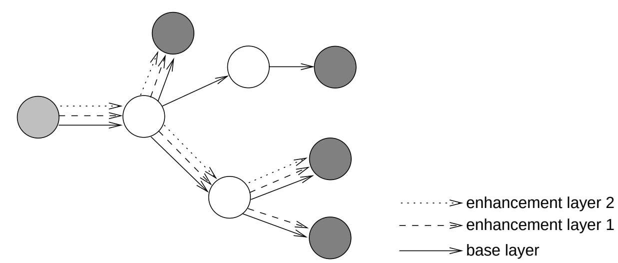


Figure 3: A multilayered multicast session showing the choice of the number of layers by the receivers according to their capacities.

The most successful multicast multilayered protocol is RLM (Receiver-driven Layered Multicast) [13]. In RLM, 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 want. 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.

It is not feasible to have one flow (for multicoding approach) or one layer (for multilayer approach) in order to satisfy different receiver capacities. Therefore, the computation of data rates in order to provide heterogeneous receivers with the fairest possible quality is a major challenge of multicast distribution [2, 5, 10, 14, 15, 16, 17].

Besides the heterogeneity of the receivers, the network must also deal with the variable characteristics of the paths from the source to the receivers. Available resources change over time and receiving capacities at the destinations cannot be considered as a static parameter. A solution to track and reduce the negative effects of variable available bandwidth is to enforce receivers to regularly inform the source about their capacities. Based on these control information, the source can compute the rates of the video layers in order to improve the video quality for the whole set of receivers. Nevertheless, the source risks to collapse if the number of members in the multicast session is too large – this is known as the feedback implosion problem.

In [15], 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. To avoid implosion of feedback control packets at intermediate nodes, it is implemented an algorithm to merge feedback packets returned by the receivers. This system will be detailed in the following.

3 Feedback implosion avoidance

A major problem for multicast applications is the feedback packet implosion at the source [18, 19]. Several implosion avoidance schemes can be found in the literature such as (pure) backoff timer, probabilistic feedback, feedback suppression, feedback aggregation/filtering, and hybrid schemes. In the backoff timer scheme, feedbacks are distributed over time, avoiding congestion at the intermediate nodes and at the source. The efficiency of such scheme depends on the chosen random delay.

For scalable multicast applications, instead of distributing the feedback messages it is mandatory to diminish the number of feedback messages. This can be achieved by the probabilistic feedback scheme, where each receiver replies to a feedback request with some probability. In this manner, the number of responses is probabilistically controlled.

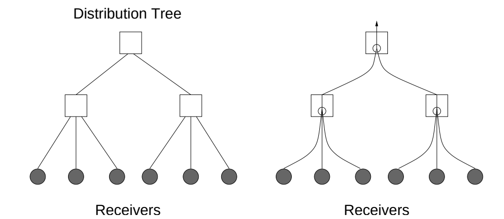
The feedback suppression scheme is commonly employed with backoff timers. In such a scheme, each receiver sets a timer to delay its feedback response from a random amount of time. The receiver must explicitly prevent the feedback generation if a feedback response from another receiver is received before the timer goes off. For scalable multicast applications requiring a tighter feedback control, the feedback suppression scheme is not well suited. As an example, consider an application where the source needs to receive periodically a feedback obtained by taking into account the feedback from the whole set of receivers. This requirement cannot be satisfied using traditional scalable feedback schemes that explicitly prevent feedback generation from many receivers. In these cases, it is recommended to use the feedback aggregation/filtering scheme where feedback messages are aggregated into a single message.

3.1 Feedback mergers

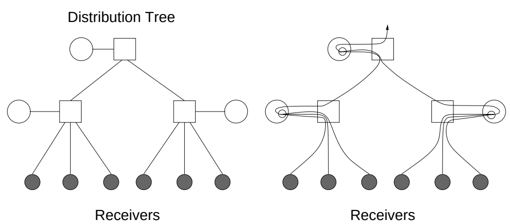
Feedback aggregation schemes can be implemented as services at the application level throughout routers as well as throughout specialized servers. Such routers or servers are referred as *feedback mergers*. Fig. 4 shows three design choices for deploying feedback mergers. In the first one, as depicted in Fig. 4(a), routers along the multicast distribution tree are capable of aggregating feedback packets. This design choice suffers from the same well-known problems in active networks. In the second design choice, as represented in Fig. 4(b), feedback mergers are co-located with routers along the multicast distribution tree. Servers that are co-located with routers are commonly employed in reliable multicast architectures. The second design choice reduces the implementation problems related to the first one while requiring minimal router support. Fig. 4(c) shows the third design choice where feedback mergers are deployed at end systems, and therefore, without requiring changes to existing routers.

3.1.1 Temporal-merging policy

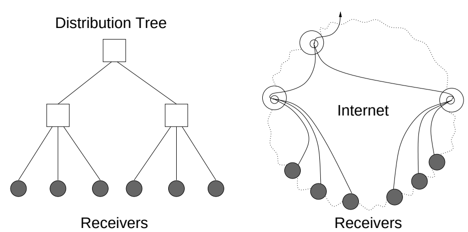
Feedback mergers employ a temporal-merging policy which determines *when* aggregating feedback packets. A temporal merging-policy commonly used by quality source-adaptive schemes is performed as follows [17, 14, 15]. Incoming feedback packets are merged and forwarded upstream if one of the following two *conditions* holds.



(a) Network Aggregation. Solution with full router support.



(b) Network Aggregation. Solution with minimal router support.



(c) End System Aggregation. No changes to existing routers.

Figure 4: Design choices to deploy feedback mergers.

1. *Packet condition:* at least one feedback packet from each node has arrived.

2. *Timeout condition:* the merger timer expires.

If more than one feedback packet from the same node arrives before merging, then the previous stored packet is replaced. As regarding this temporal-merging policy, the following two *approaches* can drive the behavior of the merger timer.

1. The merger timer expires in a periodical fashion.

2. The merger timer is triggered by the arrival of each first candidate packet for merging.

Henceforth, we present the model and the analysis of the temporal merging-policy coupled with the periodical timer expiration approach [20]. The model of the temporal merging-policy assuming that the merger timer is driven by the second approach can be regarded as a particularization of the model presented in this paper. Despite this fact, the analysis of such assumption is outside the scope of this paper.

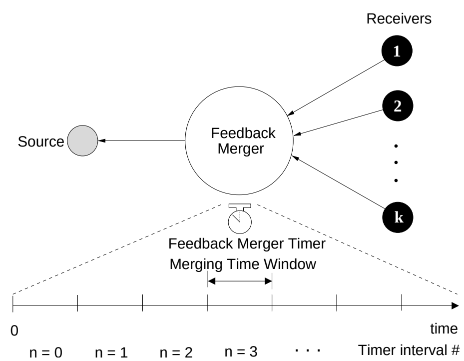


Figure 5: Basic network topology and merger timer schematics.

Consider our basic network topology as shows Fig. 5. The feedback-merger capable-node waits for packets from k receivers. Let n be a given timer interval and N a number of timer intervals. Let ρ_i^n and β_i^n be, respectively, the number of packets arrivals from node i and the number of packet replacements for node i in the timer interval n. For the periodical timer expiration approach, a timer interval n exists even if there are no feedback packets in such an interval. In this case, we consider the timer is active if $k > 1$.

The fraction of merges due to *packet* and *timeout* conditions is, respectively, given by

$$\alpha_k^{packet} = \frac{\sum_{n=0}^{N-1} \min_{1 \le i \le k} [\rho_i^n - \beta_i^n]}{\sum_{n=0}^{N-1} \max_{1 \le i \le k} [\rho_i^n - \beta_i^n]} \qquad 0 \le \alpha_k^{packet} \le 1 \qquad (1)$$

and

$$\alpha_k^{timeout} = 1 - \frac{\sum_{n=0}^{N-1} \min_{1 \le i \le k} [\rho_i^n - \beta_i^n]}{\sum_{n=0}^{N-1} \max_{1 \le i \le k} [\rho_i^n - \beta_i^n]} \qquad 0 \le \alpha_k^{timeout} \le 1 \ . \quad (2)$$

These formulæ hold if for for some $0 \le n \le (N-1)$ and $1 \le i \le k$, there exists a ρ_i^n such that $\rho_i^n \ge 1$. For an efficient packet merging, the fraction of mergers due to the *packet condition* must be maximized which implies the minimization of the fraction of mergers due to the *timeout condition*. These results are achieved when there are no packet replacements and $\rho_1^n = \rho_2^n = \ldots = \rho_k^n$ for all $0 \le n \le (N-1)$.

3.1.2 Packet filtering

The notion of a high packet filtering level is commonly employed as a measure of efficiency for implosion avoidance. In order to evaluate the packet filtering level provided by the feedback merger, we define the *Global Filtering Level*. This quantity represents the number of packets filtered by the feedback merger over the total number of incoming packets. The *Global Filtering Level* is given by

$$\xi_k = 1 - \frac{\sum_{n=0}^{N-1} \max_{1 \le i \le k} [\rho_i^n - \beta_i^n]}{\sum_{n=0}^{N-1} \sum_{i=1}^{k} \rho_i^n} \qquad \text{for } k > 1 \ , \qquad (3)$$

and if for some $0 \le n \le (N-1)$ and $1 \le i \le k$, there exists a ρ_i^n such that $\rho_i^n \ge 1$. If $k \le 1$ then $\xi_k = 0$. If feedback packets are efficiently aggregated, the *Global Filtering Level* assumes an optimal value given by

$$\xi_{k,optimal} = 1 - \frac{1}{k} \qquad \text{for } k > 1 \ . \qquad (4)$$

If there are no packet replacements, ξ_k will be less than or equal to $\xi_{k,optimal}$. If, on the other hand, there are packet replacements, ξ_k may assume values greater than $\xi_{k,optimal}$. If $\alpha_k^{timeout}$ is equal to 0 then ξ_k will be greater than or equal to $\xi_{k,optimal}$. In this case, ξ_k will be equal to $\xi_{k,optimal}$ if there are no packet replacements and it will be greater than $\xi_{k,optimal}$ if there is at least one packet replacement. Thus, despite a *Global Filtering Level* (ξ_k) greater than its optimal value impacts better the efficiency for feedback implosion avoidance, it is not adequate to an effective feedback.

Despite we analyze a network topology with a single feedback merger, the results above can be extended to any network topology with a number of feedback mergers hierarchically organized. In this manner, if in such scenario every feedback from the whole set of receivers is effectively aggregated into one feedback response, the new optimal value for the *Global Filtering Level* will be given by

$$\xi_{K,optimal} = 1 - \frac{1}{K} \qquad \text{for } K > 1, \qquad (5)$$

where K is the number of receivers in the multicast group. This result shows that feedback aggregation can effectively solve the feedback implosion problem. As an example, in a multicast group with 1,000,000 receivers, 99.99% of the feedback packets will be filtered before reaching the source.

3.1.3 Timer setting

When adopting a periodical timer expiration approach, it is possible to aggregate efficiently feedback packets. For this purpose, the receivers must send feedback in a periodical fashion and the feedback mergers must adopt a suitable merging time window. Let $\Delta\tau$ and Δt be, respectively, the feedback sending interval and the merging time window size. Supposing no packet loss and ignoring unpredictable random delays, the following relationship between $\Delta\tau$ and Δt allows an efficient aggregation.

$$\Delta\tau = b\Delta t \ , \tag{6}$$

where $b \geq 1$. The factor b can be regarded as the expected number of feedback packets from each node per timer interval. In practice, random delays and packet loss may occur. In this manner, the greater is b, the greater is the probability of packet replacements which is not adequate to an effective feedback. From this fact, b should be equal to 1. The relationship between $\Delta\tau$ and Δt imposed by Eq. (6) shows that the temporal-merging policy coupled with the periodical timer expiration approach has a low immunity to random delays, and as a consequence, feedback mergers will not be able to aggregate all feedback in an efficient manner.

4 Quality control schemes

4.1 Goodput-based approach

In the control scheme proposed in [15], every feedback control packet contains two fields: a rate π and the number of receivers, ϕ, requiring this rate. Clearly, when a packet leaves a receiver, we have $\pi = r$, where r is the receiving capacity of the receiver, and $\phi = 1$. These values are changed as this packet crosses intermediate nodes and is merged to other feedback packets. At the intermediate nodes, the arriving feedback packets are stored in a temporary array with the data rates disposed in such a way that $\pi_{e_{i-1}} \leq \pi_{e_i} \leq \pi_{e_{i+1}}$, where $(e_1, \ldots, e_{i-1}, e_i, e_{i+1}, \ldots, e_K)$, are the entries of the temporary array. 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 e_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

$$G = \sum_{k=1, k\neq j}^{K} \pi_{e_k} \phi_{e_k} \, , \tag{7}$$

where K is the number of entries in the temporary array. The number of destinations in the discarded entry is added to the number of destinations in the $(j-1)$th entry. These steps are repeated, and entries are discarded, until the number of entries in the temporary array is reduced to the maximum number of layers that the source can generate.

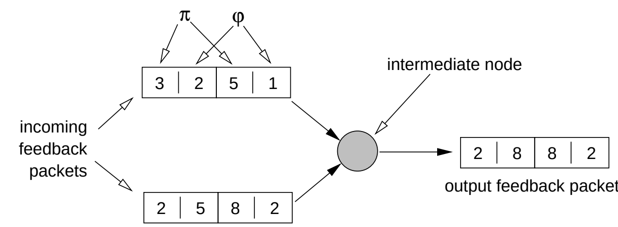


Figure 6: Example of merging procedure for the goodput-based approach.

4.2 Multimetric approach

Despite the efforts applied to improve the quality of multicast multilayered schemes, the usual "goodput-based" approaches have some limitations that arise when we consider QoS-parameters other than the global degradation at the receivers. In [17], we proposed a single function which takes into account the density of satisfied users, the amount of allocated bandwidth in the multicast tree, and the degradation at the receivers. This function allows the user to adjust the weight of each metric in such a way that the application can be classified according to one of three "pool of services." A pool of services is characterized by its correspondent prioritized metric. A specific application that aims at saving bandwidth, for instance, should give more weight to the metric that corresponds to the bandwidth allocated through the multicast tree. The same should be applied to an application that requires the optimization of the quality of the presentation at the receivers.

The classical goodput approach considers the degradation at the receivers as the unique metric when computing the layers. Let r_i be the data rate required by receiver i and $\hat{r}_i$ be the actual received rate. Thus, the degradation at receiver i is $\delta_i = r_i - \hat{r}_i$ and the *global degradation* of the multicast session is

$$\delta_{\mathbf{S}} = \sum_{i=1}^{N} \delta_i, \tag{8}$$

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

The global degradation at the receivers is certainly one of the most important parameters for defining a fair multicast multilayered session. Nevertheless, we argue in that the global degradation should not be considered as the only one. In [17], we have analyzed several aspects of transmitting multilayered applications through multicast groups based only on the global degradation at the receivers. Suppose a scenario with one source and four receivers $(r_1, \ldots, r_4)$, where the source transmits two layers, $\mathbf{L} = \{l_1, l_2\}$ Mbps, and the merging algorithm only uses the global degradation at the receivers to perform the merging procedure. Consider in a first time that $r_1 = 1$ Mbps, $r_2 = r_3 = 2$ Mbps, and $r_4 = 3.9$ Mbps. Since the rate of the base layer is limited by the slowest receiver, in this case, the global degradation at the receivers is minimized with $L = \{1, 1\}$ Mbps. Therefore, the correspondent global degradation is $\delta = 1.9$ Mbps for a bandwidth utilization of 2 Mbps.

Consider now that the capacity of receiver 4 changes from 3.9 to 4.1 Mbps. The new layers that result in the lowest global degradation are l_1=1 Mbps and l_2=3.1 Mbps, and the correspondent global degradation is δ=2 Mbps. Note that for a variation of 5% in the receiving capacity of receiver 4, layer 2 increases its global rate of $\sim$300%, which results in a substantial growth of the bandwidth utilization. Moreover, in the first scenario there were three destinations receiving 100% of the required bandwidth (and only one destination receiving $\sim$50%), while in the second case only two receivers are supplied with 100% and two receivers with 50%.

It is quite intuitive that it is better to maintain l_1 and l_2 at 1 Mbps in order to have lower bandwidth utilization even if the global degradation is slightly greater than the optimum value. The example above shows that the use of a single metric may not be always adapted to all types of applications. For example, in a pay-per-view video service, it is important that subscribers receive the maximum possible video quality even if the server rejects new calls. In video surveillance, maximizing the video quality may not be the main objective, but the reduction of the allocated network resources. On another side, broadcast advertisements may require that the data are received by a number of destinations as large as possible. We argue that the use of different criteria can reduce the problem of multicast delivery of multilayered applications if a scheme of weighted parameters is applied. Each one of the proposed metrics is prioritized depending on the application that is carried out.

We define three parameters used in our scheme: degradation at the receivers (δ), density of satisfied users (γ), weighted bandwidth utilization (ω).

As before, the first metric is the degradation at the receivers. We will not redefine this metric here. Nevertheless, it is interesting to note that the degradation at the receivers behaves quite differently from the other QoS parameters, which will be defined below.

The second metric is the density of satisfied users. For a receiver i requesting r_i unities of bandwidth, we say that it is partially satisfied if it receives r_i^r, with $r_i^r < r_i$. We define the degree of satisfaction for receiver i as

$$\gamma_i = \frac{r_i^r}{r_i}. \tag{9}$$

In a multicast session with heterogeneous receivers, we estimate the global density of satisfied users by combining γ_i of each receiver.

The third metric used by our enhanced QoS system is the weighted allocated bandwidth, denoted by ω. The level of bandwidth utilization in the network is computed directly at the source by combining the rates of the layers and the number of requests for each one of them. The value of ω in the resulting merged packet is the sum of the weighted allocated bandwidth in the downstream tree, given by

$$\omega_{out} = 1 + \sum_{i=1}^{I} \omega_{p_i}, \tag{10}$$

where $p_i, 1 \leq i \leq I$, are the incoming packets, I is the number of incoming packets, and ω_{p_i} is some rate r multiplied by the number of links crossed by this rate.

The main drawback of such an approach is that the optimization of the metrics cannot be obtained at the same time. For instance, δ and

ω should be minimized to yield an improved global quality, but what we see is that δ typically grows when ω decreases, and vice-versa. As we have argued, applications have different requirements and the way the QoS metrics should be composed is particular in each case. Based on these requirements, we define as the *global quality of the multicast session* the parameter Q that is a result of mixing γ, ω, and δ. The first rule Q must follow is the direct correspondence to the metrics. As we have seen, the global quality of the multicast session must grow if:

- The density of satisfied users grows.
- The weighted bandwidth utilization decreases.
- The quality degradation decreases.

Our quality measure is the function $Q(\gamma,\omega,\delta)$. Despite some correlation between these parameters, we have considered each metric independently because of the complexity and inaccuracy that would be introduced if we had taken into account these correlations. This simplification yields a more tractable management of the metrics and does not interfere in our objective. The function Q has the following aspect

$$Q(\gamma,\omega,\delta) = f_\gamma(\gamma) + f_\omega(\omega) + f_\delta(\delta). \quad (11)$$

We define this function to be normalized between 0 and 1. Each one of the right-sided functions of Eq. (11) is also normalized between 0 and 1. Since γ, ω, and δ depend tightly on the topology of the network, we cannot define a closed function to accomplish our needs. To implement such a feature, we have to know the minimum and maximum bounds on these variables. The minimum value is zero in all three cases, which corresponds to the cases when the source is not transmitting (for γ and ω) and when all of the receivers are being provided with the required rate (for δ). The maximum values are:

$$\gamma_{max} = 1 \quad (12)$$

$$\omega_{max} = \lim_{\max[\phi_{p_i}] \to \sum_{i=1}^{K} \phi_{p_i}} \frac{\sum_{i=1}^{K} \phi_{p_i}\pi_{p_i}}{\sum_{i=1}^{K} \phi_{p_i}} = \max[\pi_{p_i}] \quad (13)$$

$$\delta_{max} = \sum_{i=1}^{K} (\phi_{p_i}\pi_{p_i} + \delta_{p_i}) \quad (14)$$

where K is the number of entries in the temporary array. The global quality of the multicast session is defined as:

$$Q(\gamma,\omega,\delta) = \frac{a\frac{\gamma}{\gamma_{max}} + b\left(1 - \frac{\omega}{\omega_{max}}\right) + c\left(1 - \frac{\delta}{\delta_{max}}\right)}{3(a+b+c)} \quad (15)$$

where the variables a, b, and c represent the weight of each metric. By adjusting these values, we can place the application in the most adequate pool of service.

Let us now perform a simple simulation where the weighted allocated bandwidth is the prioritized metric. We have analyzed the methodology over a multicast session composed by one source and ten receivers. We use symmetric links and homogeneous delays and capacities in such a way to force the results to depend only on the available bandwidth. Variable resources are obtained in the links by introducing a background traffic generated by foreigner exponential sources. The available bandwidth is obtained by the difference between the link capacity and the background traffic. Fig. 7 shows the results for the proposed algorithm based on three metrics and the goodput-based approach. After a temporary stabilization period, the proposed approach results in reduced allocated bandwidth. When compared with the classical approach the new approach leads to 30% of bandwidth saving.

This curve depicts the behavior of the system when a large number of merging procedures are performed in one node. This shows that the system responds well to heterogeneous receivers and also that the quality of the multicast session is quite stable during the communication.

5 Cooperative receivers

The feedback merging approach supposes some kind of "activity" in the routers, which must capable of manipulating the packets's contents. Nevertheless, this may not be valuable for some architectures such as the current Internet. In [15], a slightly different version of this concept

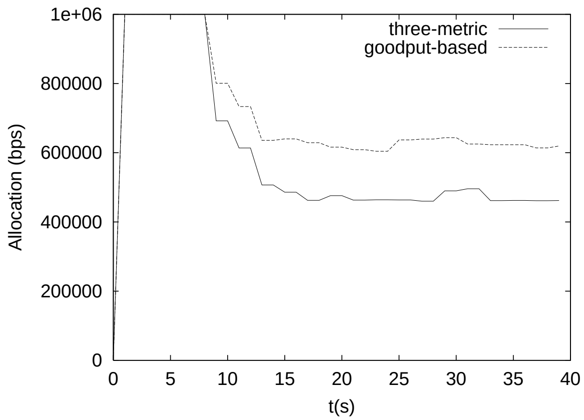


Figure 7: Allocated bandwidth through the multicast tree.

is used. Some nodes in the network are designed to function as feedback mergers and aggregate packets that are sent intentionally to them. Thus, the problem of implementing extra functionalities in the totality of the routers is reduced to a subset of the nodes. Nevertheless, the problem is only rescaled (but still existing) and the network must now find which entities should be designated as feedback mergers. Indeed, current architectures like the Internet are not prepared yet to promptly accommodate such functionalities.

A solution to this problem is to implement feedback suppression at the receivers. In such an approach, based on some pre-established rules, some receivers cancel their feedbacks before they are sent to the source. This approach can be implemented incurring minor changes in the current Internet architecture – this require only the IP-Multicast protocol. Another solution would be to use application-level multicast. In this approach, the root, the leaves, and the intermediate nodes are all members of the multicast tree. Feedback mergers can then be implemented at end systems, without requiring any changes in the network infrastructure.

We define a system that improves the cooperation between the source and the receivers in order to improve the global video quality for the whole set of receivers. The mechanism is started by the receivers upon the reception of a prospection packet. The prospection packet is sent regularly by the source to the receivers in order to trigger the generation of feedback control packets. Thus, the arrival of the prospection packet p_i at receiver i starts a sequence of verification procedures and message exchange among receivers.

Let t_i be the time that the prospection packet arrives at receiver i. We consider that all prospection packets arrive at the receivers at the same time. We call this time t and thus $t = t_{i=1} = t_{i=2} = \ldots = t_{i=(N-1)} = t_{i=N}$, where N is the number of members in the multicast session.

Let also τ_i be the timer length for receiver i and $t_{i,j}$ be the transmission delay between receivers i and j. We suppose the network to be symmetric, so $t_{i,j} = t_{j,i}$.

Receiver i configures its timer as soon as it receives the prospection packet. When timer τ_i goes off, receiver i multicasts an acknowledgment, p_i, to the source and to the other members. This ACK will induce the execution of the suppression algorithm by receiver j only if

$$\tau_i + t_{i,j} < \tau_j. \tag{16}$$

In order to implement such mechanisms, we must build a special prospection packet. This packet must contain an entry correspondent to the *accumulation index* α. This index is the threshold that determines if a receiver will be "aggregated" or not to another one. The aggregation process respects the following definition.

Definition 1 *A receiver i asking for r_i units of bandwidth and waiting for its timer to go off suppresses its ACK if, upon the arrival of an ACK from another receiver j, the inequality $\frac{r_j}{r_i} \geq \alpha$ is respected.*

As a consequence, the number of acknowledgments that arrive at the source is also controlled. The accumulation index can also be dynamically adjusted by the source in order to maintain the density of feedbacks within an acceptable value (previously determined, e.g. equivalent to 5% of the number of receivers).

We have simulated our approach in order to observe whether the system leads to good suppression levels. We are mainly interested in the scalability of the system when varying the average timer (T), the suppression index (α), and the transmission delay between receivers (d).

The simulation scenario is composed of one source and N receivers. In a first time, we consider $N = 10$. All configurations suppose the same inter-receiver delay $d = 10$ms. Timers are uniform with parameter T. It is more interesting to set timers as multiple of the transmission delay between receivers. We consider $T = 2d$, $T = 8d$, and $T = 15d$. To represent the heterogeneity among receivers, we use background traffics following an exponential law. The receiving capacities vary in the range from 100 Kbps to 1 Mbps.

Fig. 8 shows the results for $\alpha = 0.9$. This corresponds to a situation where receivers are low tolerant. For short timers ($T = 2d$), the number of suppressions is about 20%. For longer timers ($T = 15d$), the number of ack suppressions is a little bit more expressive, about 50%.

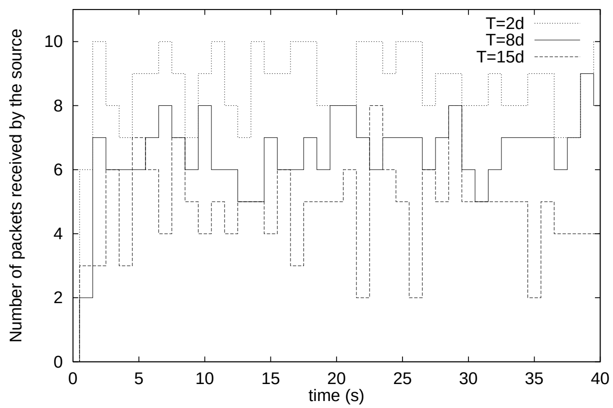


Figure 8: Total number of feedback packets received by the source for $\alpha = 0.9$.

We evaluate now a more tolerant scenario, where the source sets the suppression index to $\alpha = 0.5$. We expect the average number of received feedback packets to drop. This result is shown in Fig. 9. We should note however that there is a tradeoff between the suppression index and the global video quality. As the suppression index increases, the probability that receiver r_i suppress its acknowledgment also increases. As a consequence, the quantity of feedback packets that arrive at the source decreases. Since the source will dispose of less information from the receivers, the computation of the video layers will probably lead to higher global video degradation.

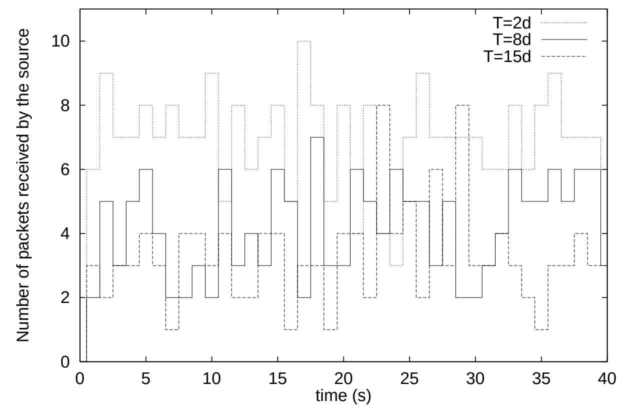


Figure 9: Total number of feedback packets received by the source for $\alpha = 0.5$.

6 Conclusion

The need for meeting QoS requirements of multicast video applications imposes new different challenges. In this paper, we have addressed mechanisms that allow feedback scalability and quality control at the source. We have shown that the choice of the implosion avoidance mechanism depends on the application requirements. There is no single solution that can satisfy all application needs. In particular, we have shown that feedback aggregation schemes can effectively solve the feedback implosion problem while allowing feedback with novel semantics to be received by the source. We have also evaluated a cooperative approach among receivers that allow to improve the global video quality for the multicast session and reduces the feedback implosion probability. We have argued that the classical goodput approach to compute the rates of the layers for source-adaptive schemes fails in the computation of the most adequate rates. We have shown that the use of a new combined metric that allows for the density of satisfied users, the wighted bandwidth allocation, and the goodput at the receivers yields enhancements in quality/fairness of multilayered multicast sessions.

Acknowledgments

This work was supported in part by grants from CAPES/COFECUB, CNPq, and UFRJ.

References

[1] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jasmin, "Resource reSerVation Protocol (RSVP) – version 1 functional specification." Request for Comments 2205, Internet Engineering Task Force, Sept. 1997.

[2] R. Rejaie, M. Handley, and D. Estrin, "Layered quality adaptation for internet video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, Dec. 2000.

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

[4] 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.

[5] R. Rejaie, M. Handley, and D. Estrin, "Quality adaptation for congestion controlled video playback over the Internet," in *ACM Sigcomm*, (Cambridge, MA, USA), Sept. 1999.

[6] 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.

[7] A. Ziviani, B. Wolfinger, J. F. de Rezende, O. C. M. B. Duarte, and S. Fdida, "On the combined adoption of QoS schemes to improve the delivery quality of MPEG video streams," in *Spects*, (San Diego, USA), July 2002.

[8] 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), Aug. 2001.

[9] T. Turletti and C. Huitema, "Videoconferencing on the internet," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 340–351, June 1996.

[10] J. C. Bolot and T. Turletti, "Experience with control mechanisms for packet video in the internet," *ACM Computer Communication Review*, vol. 28, no. 1, Jan. 1998.

[11] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications." Request for Comments 1889, Internet Engineering Task Force, Jan. 1996.

[12] T. Jiang, M. Ammar, and E. W. Zegura, "On the use of destination set grouping to improve inter-receiver fairness for multicast ABR sessions," in *IEEE Infocom*, (Tel-Aviv, Israel), Mar. 2000.

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

[14] M. D. de Amorim, O. C. M. B. Duarte, and G. Pujolle, "Improving user satisfaction in adaptive multicast video." to appear in *Journal of Communications and Networks*, 2002.

[15] B. J. Vickers, C. Albuquerque, and T. Suda, "Source adaptive multi-layered multicast algorithms for real-time video distribution," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 720–733, Dec. 2000.

[16] J. Byers, M. Luby, and M. Mitzenmacher, "Fine-grained layered multicast," in *IEEE Infocom*, (Anchorage, Alaska), Apr. 2001.

[17] M. D. de Amorim, O. C. M. B. Duarte, and G. Pujolle, "Distinguishing video quality through differential matrices." to appear in *ACM Multimedia Systems Journal*, 2003.

[18] T. Fuhrmann and J. Widmer, "On the scaling of feedback algorithms for very large multicast groups," *Computer Communications*, vol. 24, no. 5–6, pp. 539–547, Mar. 2001.

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

[20] P. A. da Silva 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.

Paulo André da S. Gonçalves received the B.Sc. degree in Electronics Engineering from the Federal University of Rio de Janeiro (UFRJ), Brazil, in 1998, and the M.Sc. degree in Electrical Engineering from COPPE/UFRJ, Brazil, in 2000. He is presently a Ph.D. student at the LIP6 Laboratory of the Université Pierre et Marie Curie - Paris 6, France. His research interests include ad-hoc mobile wireless networks, multicast, and video distribution.

Marcelo D. de Amorim received the "Cum Laude" Degree in Electronics 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), France. He is an external collaborator of the Networking and Automation Group (GTA) of COPPE/UFRJ. His major research interests are in QoS guarantees, mobility, peer-to-peer networks, and large-scale routing.

José F. de Rezende received the B.Sc. and M.Sc. degrees in Eletronical Engineering from the Federal University of Rio de Janeiro (UFRJ) in 1988 and 1991, respectively. He received the Ph.D. degree in Computer Science from the Université Pierre et Marie Curie, Paris France, in 1997. He was an associate researcher at LIP6 (Laboratoire d'Informatique de Paris 6) during 1997. Since 1998 he is an Associate Professor at UFRJ. His research interests are in distributed multimedia applications, multipeer communication, high speed and mobile networks and quality of service in the Internet.

Otto Carlos M. B. Duarte received the Electronic Engineer degree and the M. Sc. degree in Electrical Engineering from the Federal University of Rio de Janeiro (UFRJ), Brazil, in 1976 and 1981, respectively, and the Dr. Ing. degree from ENST/Paris, France, in 1985. Since 1978 he is Associate Professor at UFRJ. From January 1992 to June 1993 he has worked at MASI Laboratory in Paris 6 University. In 1995, he has spent three months at International Computer Science Institute (ICSI) associated to the University of California at Berkeley. In 1999 and 2001, he worked as invited professor at Paris 6 University. He is heading the computer network group (Grupo de Teleinformtica e Automao - GTA). His major research interests are in multicast, QoS garantees, security and mobile communications.