Improving Feedback Merging for Source-Adaptive Layered Multicast Schemes *

Paulo André da S. Gonçalves[1,2,†], (paulo.goncalves@lip6.fr) José F. de Rezende[2] (rezende@gta.ufrj.br), Otto Carlos M. B. Duarte[2] (otto@gta.ufrj.br) and Guy Pujolle[1] (guy.pujolle@lip6.fr)

[1] *Laboratoire d'Informatique de Paris 6 (LIP6)*
Université Pierre et Marie Curie (UPMC)
8, rue du Capitaine Scott
75015 - Paris - France
Phone: +33 1 44 27 61 88
Fax: +33 1 44 27 74 95

[2] *Grupo de Teleinformática e Automação (GTA)*
COPPE/EE – Universidade Federal do Rio de Janeiro (UFRJ)
P.O. Box 68504 – 21945-970
Rio de Janeiro - RJ - Brazil
Phone: +55 21 2260-5010 Ext. 240
Fax: +55 21 2290-6626

March 21, 2003

Abstract. Current source-adaptive layered multicast schemes exploit merging capabilities at special nodes in the network to combine feedback from the multicast group. The merging procedures reduce network load and avoid feedback implosion at the source. In this paper, we demonstrate how to increase the efficiency of a merger node. We show how to tune two mechanisms: "feedback transmission", on the timer settings used by receivers, and "temporal merging", on the timer settings used by intermediate nodes. Our approach is generic, relating purely to timer settings, and does not touch on "content merging", or the decision regarding which packets to forward. We investigate the effects of temporal merging on packet suppression. We show that periodic feedback transmission coupled with a suitable timer setting at the merger node increases efficiency.

Keywords: feedback merging, performance, multicast

1. Introduction

The need for meeting the Quality of Service (QoS) requirements of multimedia applications is steadily increasing. As a consequence, new approaches, such as adaptive applications [9], QoS routing [20, 12, 5, 4], and Differentiated Services [3, 17, 22] are currently being developed.

* This work was sponsored by CAPES/COFECUB, FUJB, CRNS, and CNPq.
† Contact author. Mailing Address: LIP6.

In the case of adaptive applications, different schemes are emerging to improve quality and fairness of layered video applications [13, 11, 15, 16, 8, 18, 6, 7, 19]. In the context of unicast, for example, Rejaie *et al.* [15, 16] propose an adaptation scheme that is robust even in the presence of fluctuations in the available bandwidth. To achieve this robustness, the receiver plays an important role by buffering video layers and sending feedback packets containing current network congestion information to the source.

Exchanging network congestion information or, more generally, any kind of information between the multicast group and the source becomes more difficult. The amount of potential feedback increases linearly with the number of receivers and may lead to feedback implosion at the source. To solve this problem, various multicast schemes [21, 10, 14] incorporate feedback trees in which internal nodes are responsible for merging the information contained in feedback packets. This merging consists of receiving information from numerous downstream nodes and then either selecting from or summarizing this information for retransmission upstream. Current source-adaptive layered multicast schemes also rely on merger nodes to combine feedback from the multicast group.

This paper is concerned with the efficiency of merger nodes. A merger node is inefficient if it generates upstream feedback before having received information from all of its downstream neighbors. It is also inefficient if it receives information from a downstream neighbor that it does not employ when generating upstream feedback (for instance, when it has had to discard downstream information because of memory limitations). We define efficiency formally in Section 4. In this paper, we demonstrate how to increase the efficiency of a merger node for current source-adaptive layered multicast schemes. We identify two factors that affect efficiency: feedback transmission and temporal merging. Feedback transmission describes the manner in which a receiver decides when to send feedback packets upstream. On the other hand, temporal merging is the manner in which intermediate nodes decide when information is to be forwarded.

The quality of a merger scheme is also dependent upon the manner in which information is chosen or amalgamated. This paper does not address this "content merging" problem, whose solutions depend to a great extent on the particular information being merged. Our work is restricted to the generic, content-neutral questions related to scheduling the flow of information. The efficiency of feedback transmission and temporal merging directly affect the quality of any content merging scheme.

We also investigate the effects of temporal merging on packet suppression. The effectiveness of implosion avoidance is commonly measured by the degree of packet suppression. The main contributions of this paper are: (1) a temporal merging model that makes no particular assumptions regarding the amount of feedback packets transmitted; (2) the description and analysis of two timer setting schemes at the merger node; (3) an analysis of the effects of temporal merging on packet suppression; and (4) a demonstration of improved merging efficiency by using periodic feedback transmission together with a suitable timer setting at the merger node.

The paper is structured as follows: Section 2 presents related work. Section 3 describes temporal merging. The formulæ that model temporal merging and an analysis of the necessary conditions to ensure efficient merging are presented in Section 4. Thereafter, the effects of temporal merging on packet suppression are examined. Section 5 presents a feedback transmission mechanism that together with temporal merging improves merging efficiency. In Section 6, we determine suitable window sizes for improved merging efficiency. Simulation results are presented in Section 7. Finally Section 8 concludes the paper.

2. Related Work

McCanne *et al.* [13] pioneered the study of multicast layered video transmission. They propose the transmission of video layers on separate multicast addresses. Based on congestion control information, a receiver subscribes to the number of layers that can be supported by its individual path from the source. Li *et al.* [11] propose a hierarchical rate control algorithm to manage the adding and dropping of video layers by receivers and an error recovery mechanism using retransmissions to adapt to network congestion. One limitation of these schemes is that receivers are limited to the layers that the source is willing to transmit. Current layered schemes rely on feedback from the multicast group to perform quality adaptation at the source.

Vickers *et al.* [18] propose a source-adaptive layered multicast algorithm in which the transmission of each receiver report is triggered by the reception of a control packet sent by the source. Control packets are transmitted once for every R video packets transmitted, where R is assumed to be a small number such as 15. This assumption in practice leads to excessive control traffic. Intermediate nodes perform content merging to estimate the rate of the video layers. This content merging is based on a combined "goodput" of video traffic received by all the receivers. Neither analysis nor results regarding the merging efficiency of intermediate nodes are provided. Also, no justifications about the choice of the tunable parameter R and the timer setting at the merger nodes are presented.

The single-loop packet merging scheme proposed in [1] only looks at the content-merging problem. Feedback transmission and temporal merging are similar to the approach in [18]. However, the analysis is based on the assumption that a merger node performs each merging procedure using one report from each of the its downstream neighbors. The proposed scheme for adapting the quality of layered video shows improvements in the fairness of the delivered service. In [7] it is demonstrated that content merging based solely on the proposed "goodput" quality metric in [18] fails to compute the best set of rates for the layers. In [8] is proposed a content-merging algorithm that considers the density of satisfied users, the weighted bandwidth allocation, and the "goodput" to the receivers. The results show enhancements in the quality and fairness of layered multicast sessions. Feedback transmission and temporal merging are also similar to the approach in [18] but for analysis purposes, each merging procedure is assumed to use one report from each of the downstream neighbors of a merger node.

In general, the above-mentioned proposals for adapting video quality at the source rely on similar temporal merging procedures. However, no attention is given to merging efficiency nor to the effects of temporal merging on feedback suppression. Our examination begins with a description of the temporal merging employed by the above-mentioned works.

3. Temporal Merging

Consider the basic network topology as shown in Figure 1. The merger node waits for feedback packets from k receivers. Temporal merging is performed as follows: incoming feedback packets are merged and the resulting packet is sent upstream whenever one of the subsequent two *conditions* holds. *(i)* At least one feedback packet from each of the nodes has arrived. *(ii)* The merger timer has expired. If, prior to merging, more than one feedback packet from the same node arrives, then the previously stored packet is replaced.

The rationales for *conditions (i)* and *(ii)* are as follows: *condition (i)* ensures that the content-merging algorithm is aware of feedback from each of the k nodes, satisfying a notion of fairness. *Condition (ii)* relates to the maximum waiting time required for *condition (i)* to hold.

Packet replacements favour the most recent received information. However, they entail wasted resources. Information previously trans-

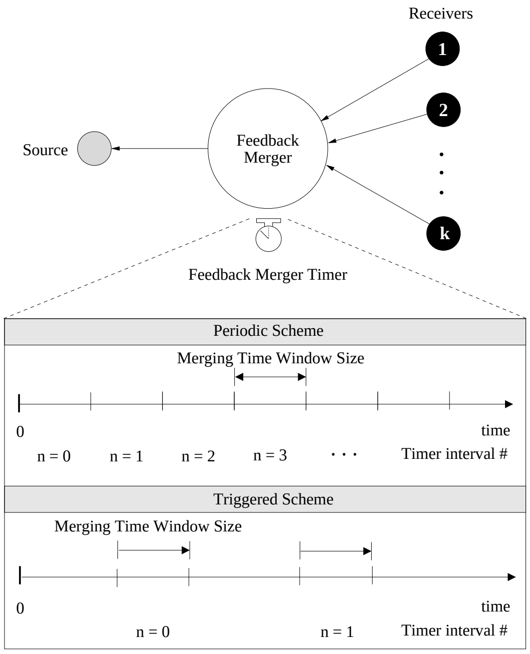

Figure 1. Basic network topology and merger timer schematics.

mitted is discarded. To merger in an efficient manner, we should find a means of reducing both the number of packet replacements and the number of merges due to *condition (ii)*. Ultimately, this problem consists of finding a feedback transmission mechanism that, coupled with suitable timer settings at the merger node, improves merging efficiency. It is possible that merger nodes do not receive any report for a long period of time. In this case, merger nodes must employ an additional timer to trigger the transmission upstream of a feedback packet, thus preventing feedback starvation at the source.

4. Temporal Merging Model

We study two schemes for regulating the behavior of the merger timer. *(i)* The merger timer expires at regular prescheduled intervals. We call this the "periodic scheme". *(ii)* The merger timer is set upon arrival of the first feedback packet following a merge. We call this the "triggered scheme". The former is simpler to implement than the latter. We will model and analyze both schemes. We will describe and analyze the formulæ to compute merges attributable to both conditions, and then analyze the effects of temporal merging on feedback suppression. Throughout this section, the variables marked with a circumflex are related to the triggered scheme. The model we present makes no particular assumptions about the amount of feedback packets transmitted.

4.1. Merges due to *conditions (i)* and *(ii)*

Consider the merger timer schematics shown in Figure 1. Let k be the number of receivers from which the feedback merger is waiting for packets. For a given timer interval n, let ρ_i^n and β_i^n be, respectively, the number of packet arrivals from receiver i and the number of its packets that have been replaced.

4.1.1. *Periodic Scheme*

Under the periodic scheme, a timer interval n occurs even in the absence of feedback. For a feedback merger waiting for packets from k receivers, the number of outgoing packets from the timer interval n is given by

$$\gamma_{k,out}^n = \max_{i=1,\ldots,k} [\rho_i^n - \beta_i^n], \qquad \gamma_{k,out}^n \geq 0, \tag{1}$$

and the number of incoming packets in the timer interval n is

$$\gamma_{k,in}^n = \sum_{i=1}^{k} \rho_i^n, \qquad \gamma_{k,in}^n \geq 0. \tag{2}$$

Given a timer interval n, the number of outgoing packets due to *condition (i)* is given by

$$\mu_k^n = \min_{i=1,\ldots,k} [\rho_i^n - \beta_i^n], \qquad \mu_k^n \geq 0, \tag{3}$$

and the number of outgoing packets attributable to *condition (ii)* is

$$\sigma_k^n = \max_{i=1,\ldots,k} [\rho_i^n - \beta_i^n] - \min_{i=1,\ldots,k} [\rho_i^n - \beta_i^n], \qquad \sigma_k^n = 0 \text{ } or \text{ } 1. \tag{4}$$

If timer expiration occurs and there exists at least one feedback packet waiting for merging, then $\sigma_k^n = 1$ else $\sigma_k^n = 0$. Note that the number of outgoing packets from a timer interval n is equal to $\mu_k^n + \sigma_k^n$, the same result as in (1). To achieve efficient merging in a timer interval n, the following two conditions must hold: $\beta_i^n = 0$ for all $1 \leq i \leq k$, and $\sigma_k^n = 0$. According to Eq. (4), it follows that if β_i^n is equal to zero for all $1 \leq i \leq k$, σ_k^n will also be equal to zero if $\max_{i=1,\ldots,k}[\rho_i^n] = \min_{i=1,\ldots,k}[\rho_i^n]$, which implies $\rho_1^n = \rho_2^n = \ldots = \rho_k^n$. Because no ρ_i^n is allowed to be equal to zero in pursuing efficient merging, $\rho_1^n = \rho_2^n = \ldots = \rho_k^n \geq 1$. It should be noted that $\rho_1^n = \rho_2^n = \ldots = \rho_k^n = 1$ means that $\beta_1^n = \beta_2^n = \ldots = \beta_k^n = 0$.

However, $\rho_1^n = \rho_2^n = \ldots = \rho_k^n > 1$ does not necessarily mean that $\beta_1^n = \beta_2^n = \ldots = \beta_k^n = 0$. Merging in a timer interval n is efficient if there are no packet replacements and the number of packet arrivals from all receivers is the same. The periodic scheme allows for more than one merging procedure attributable to *condition (i)* per timer interval. The number depends upon timer settings and feedback transmission characteristics.

Considering N timer intervals, the fraction of merges due to *condition (i)* is given by

$$\alpha_k^{cond(i)} = \frac{\sum_{n=0}^{N-1} \min_{i=1,\ldots,k}[\rho_i^n - \beta_i^n]}{\sum_{n=0}^{N-1} \max_{i=1,\ldots,k}[\rho_i^n - \beta_i^n]}, \qquad 0 \leq \alpha_k^{cond(i)} \leq 1, \tag{5}$$

and the fraction of merges attributable to *condition (ii)* is

$$\alpha_k^{cond(ii)} = 1 - \frac{\sum_{n=0}^{N-1} \min_{i=1,\ldots,k}[\rho_i^n - \beta_i^n]}{\sum_{n=0}^{N-1} \max_{i=1,\ldots,k}[\rho_i^n - \beta_i^n]}, \qquad 0 \leq \alpha_k^{cond(ii)} \leq 1. \tag{6}$$

The above-mentioned formulæ hold if for some n, $0 \leq n \leq (N-1)$, there exists a i, $1 \leq i \leq k$, such that $\rho_i^n \geq 1$.

4.1.2. *Triggered Scheme*

In the triggered scheme, at least one feedback packet is necessary in order to start a timer interval. It is easy to argue that for a feedback merger waiting for packets from k receivers, the number of outgoing packets from a timer interval n is $\hat{\gamma}_{k,out}^n = \max_{i=1,\ldots,k}[\rho_i^n - \beta_i^n] = 1$. To see this, consider that the arrival of a feedback packet pertaining to a receiver j sets the merger timer. For such a receiver, $\rho_j^n - \beta_j^n$ is equal to one because a timer interval will be finished after the occurrence of a merging procedure. By way of the same reasoning, it can be easily verified that for a receiver i, where $i \neq j$, $\rho_i^n - \beta_i^n$ is at the most one.

A simplification of the presentation of the triggered scheme model will be made. We will only present the formulæ that are pertinent to the analysis throughout this paper. Given a timer interval n, the number of outgoing packets due to *condition (ii)* is given by

$$\hat{\sigma}_k^n = 1 - \min_{1\leq i\leq k}[\rho_i^n - \beta_i^n] \ , \qquad \hat{\sigma}_k^n = 0 \ or \ 1. \tag{7}$$

It should be noted here that in a timer interval n, if β_i^n is equal to zero for all $1 \leq i \leq k$, then $\hat{\sigma}_k^n$ will be equal to zero if $\min_{1\leq i\leq k}[\rho_i^n]$ is equal to one. That is, $\rho_1^n = \rho_2^n = \ldots = \rho_k^n = 1$. Thus, regardless of the amount of transmitted feedback packets, the model shows that an efficient merging in a timer interval n is achieved if the number of packet arrivals from all receivers is equal to one.

Considering N *existing* timer intervals, the fraction of merges attributable to *condition (i)* is given by

$$\hat{\alpha}_k^{cond(i)} = \frac{1}{N} \sum_{n=0}^{N-1} \min_{1 \le i \le k} [\rho_i^n - \beta_i^n] \ , \qquad 0 \le \hat{\alpha}_k^{cond(i)} \le 1, \tag{8}$$

and the fraction of merges due to *condition (ii)* is

$$\hat{\alpha}_k^{cond(ii)} = 1 - \frac{1}{N} \sum_{n=0}^{N-1} \min_{1 \le i \le k} [\rho_i^n - \beta_i^n] \ , \qquad 0 \le \hat{\alpha}_k^{cond(ii)} \le 1 \ . \tag{9}$$

4.2. Feedback Suppression

The effectiveness of implosion avoidance is commonly measured by the degree of packet suppression. This section investigates the effects of temporal merging on packet suppression.

4.2.1. *Periodic Scheme*

We define *Filtering Level* (ξ^n) in a timer interval n as the number of packets suppressed by the feedback merger over the total number of incoming packets, i.e.,

$$\xi^n = 1 - \frac{\gamma_{out}^n}{\gamma_{in}^n}, \qquad 0 \le \xi^n < 1, \tag{10}$$

if $\gamma_{in}^n \ge 1$. In generalizing our definition to take into account a number N of timer intervals, we get the *Global Filtering Level*

$$\xi_k = 1 - \frac{\sum_{n=0}^{N-1} \gamma_{out}^n}{\sum_{n=0}^{N-1} \gamma_{in}^n}, \qquad 0 \le \xi_k < 1. \tag{11}$$

Substituting equations (1) and (2) in (11), we obtain

$$\xi_k = 1 - \frac{\sum_{n=0}^{N-1} \max_{i=1,\dots,k} [\rho_i^n - \beta_i^n]}{\sum_{n=0}^{N-1} \sum_{i=1}^{k} \rho_i^n}, \qquad \text{for } k > 1, \tag{12}$$

and if for some n, $0 \le n \le (N-1)$, there exists a i, $1 \le i \le k$, such that $\rho_i^n \ge 1$. If $k \le 1$ then $\xi_k = 0$.

We can now derive the *optimal value* of ξ_k, which we define as the value that results from efficient merging. For this purpose, consider both the network example as depicted in Figure 1 and the following scenario in which merging is efficient. Assume that for every timer interval n having packets, there are no packet replacements and the number of packet arrivals from all receivers is the same. So that, for any timer interval n having packets, the $\frac{\gamma_{out}^n}{\gamma_{in}^n}$ ratio will always be constant and equal to $\frac{1}{k}$. In the same vein, the $\frac{\sum_{n=0}^{N-1}\gamma_{out}^n}{\sum_{n=0}^{N-1}\gamma_{in}^n}$ ratio will also be equal to $\frac{1}{k}$. In this manner, the *optimal value* of ξ_k is given by

$$\xi_{k,optimal} = 1 - \frac{1}{k}, \qquad \text{for } k > 1. \tag{13}$$

4.2.2. *Triggered Scheme*

Under the triggered scheme, the Global Filtering Level is given by

$$\hat{\xi}_k = 1 - \frac{N}{\sum_{n=0}^{N-1}\sum_{i=1}^{k}\rho_i^n}, \qquad 0 \leq \hat{\xi}_k < 1, \tag{14}$$

for N existing timer intervals; and the optimal value of $\hat{\xi}_k$ is

$$\hat{\xi}_{k,optimal} = \xi_{k,optimal}. \tag{15}$$

4.2.3. *Remarks*

The following remarks hold for both timer schemes. The Global Filtering Level should be as close as possible to its optimal value in pursuit of efficient merging. However, an optimal value of the Global Filtering Level does not imply efficient merging because packet replacements and merges due to *condition (ii)* may be occurring. Figure 2 illustrates this situation when the periodic scheme is used. In relation to the timer intervals 2, 3, and 4, the Global Filtering Level (ξ_3) is equal to its optimal value of 0.66.

Other important remarks follow: if there are no packet replacements, the Global Filtering Level will be less than or equal to its optimal value. If, on the other hand, there are packet replacements, the Global Filtering Level may assume values greater than its optimal value. If the fraction of merges due to *condition (ii)* is equal to zero, then the Global Filtering Level will be greater than or equal to its optimal value. In this case, the Global Filtering Level will be equal to its optimal value if there are no packet replacements; and it will be greater than its optimal value if there is at least one packet replacement. Thus, a Global Filtering Level that is greater than its optimal value increases efficiency for feedback implosion avoidance, but is inadequate for effective feedback.

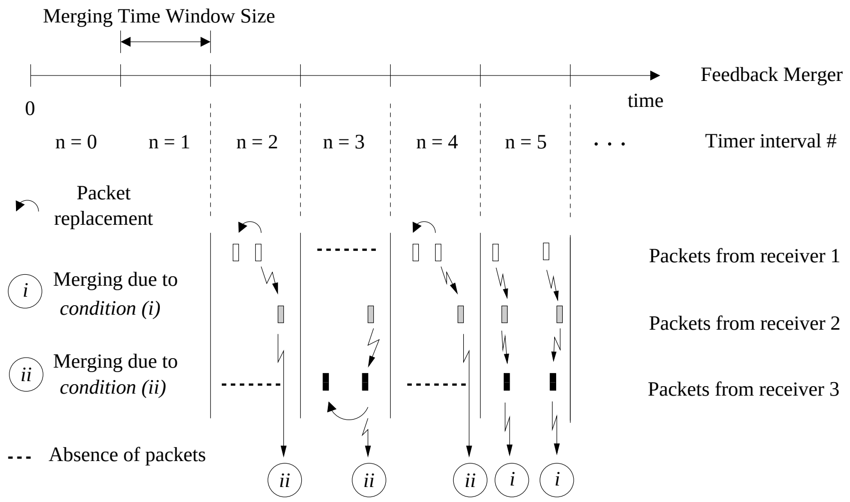

Figure 2. Scenario in which the Global Filtering Level assumes its optimal value.

5. Feedback Transmission

An optimal solution for transmitting feedback should share the session bandwidth devoted to the feedback traffic equally among the receivers. In this case, all receivers must issue reports at the same rate. However, transmission of receiver reports need to be coordinated to achieve efficient merging. One solution to this problem is the employment of a feedback polling mechanism on the part of feedback mergers. Thus, each merger node will be able to receive reports or merged reports in a coordinated fashion.

Due to packet loss, not all recipients will receive a request for a report, potentially degrading merging efficiency. In general, schemes that explicitly coordinate the transmissions of all receiver reports tend to generate excessive control traffic. In addition, determining an optimal timer setting at the merger nodes may be difficult.

In this paper, we adopt a feedback transmission mechanism whereby a steady state leads to a reasonably efficient merging capability. This mechanism relates purely to the timer settings used by receivers, and thus avoids the above-mentioned problem. Feedback transmission proceeds as follows: each receiver regularly transmits consecutive feedback packets spaced apart by Δt. The coordination of the packet arrivals at the merger node is dependent on the amount of jitter experienced by each receiver's feedback stream. We determine a lower bound on the feedback sending interval that reduces the effect of jitter and a suitable merging time window size.

6. Window Size and Lower Bound on Δt

We approach the problem of determining a lower bound on the feedback sending interval and a suitable merging time window by analyzing two scenarios: one that disregards random delays and one that takes them into account. Consider our basic network topology (Figure 1). Each receiver regularly sends consecutive feedback packets spaced apart by Δt. The merging time window size is $\Delta \tau$. We assume that the feedback sending rate is supported on each receiver-feedback merger path. All feedback packets have the same fixed size [18, 1, 7, 8].

6.1. First Scenario: Disregarding Random Delays

In addition to the inexistence of random delays, we also assume that there is no packet loss. Thus, packet replacements cannot occur.

6.1.1. *Periodic Scheme*

For $k = 2$ and $k = 3$, Figure 3 shows steady-state values of ξ_k and $\alpha_k^{cond(ii)}$ as a function of the feedback sending interval. The plot is a scatter plot. For each Δt, different values of ξ_k and $\alpha_k^{cond(ii)}$ are obtained by varying the arrival-times of all receivers' first reports over the interval $[0, \Delta t)$. By steady-state values, we mean values computed over a recurring pattern of merges at the merger node. The main conclusions regarding these figures are the following. First, merges due to *condition (ii)* lead to a Global Filtering Level that is lower than its optimal value. Second, setting the merging time window size for a multiple of the feedback sending interval ensures that all merges occur due to *condition (i)*. That is, assuming that there is no packet loss, efficient merging is achieved if $\Delta\tau = b\Delta t$, where $b = 1, 2, \ldots$. Factor b represents the expected number of feedback packets from each receiver per timer interval. One method of reducing the probability of packet replacements in the presence of packet loss is to make $b = 1$. This choice also reduces the maximum possible waiting time for merging. Hence, we adopt $b = 1$.

Finally, it should be noted that if the feedback sending interval Δt (or the feedback interarrival interval) is around window size $\Delta\tau$, then the performance of the periodic scheme may decline. The occurrence of random network jitter leads to a feedback interarrival interval close to the merging time window size. We should expect, therefore, the periodic scheme to perform well only when the feedback sending interval value is higher than the value of jitter. How much higher the feedback sending interval value should be is investigated in Section 7.

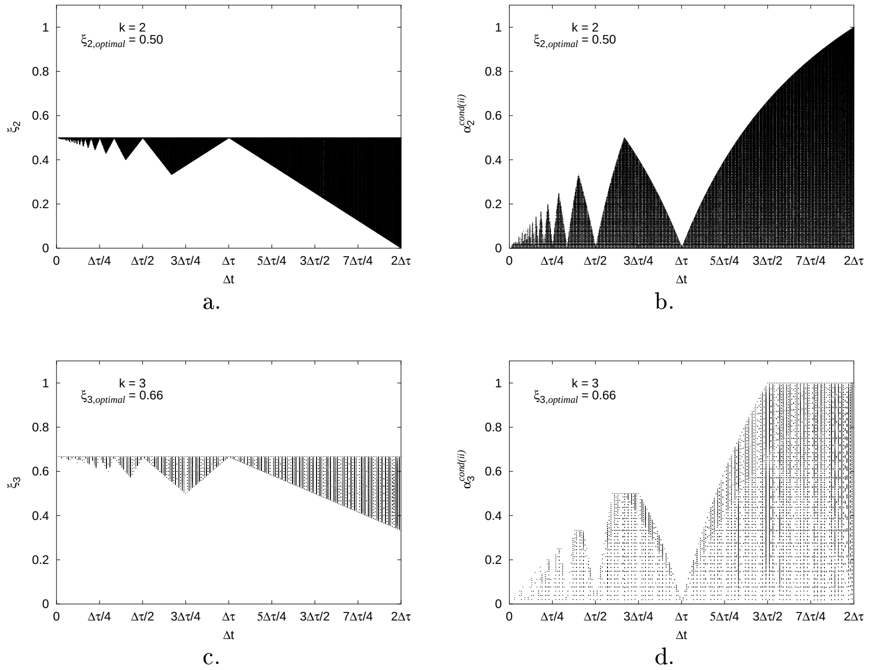

Figure 3. Theoretical results for the periodic scheme.

6.1.2. *Triggered Scheme*

For $k = 2$, $k = 3$, and $k = 4$, Figure 4 shows the steady-state values of $\hat{\xi}_k$ and $\hat{\alpha}_k^{cond(ii)}$ as function of the feedback sending interval. See Section 6.1.1 for the definition of steady-state values and details about the arrival-times of receivers' first reports. Note that, as highlighted in each plot, efficient merging is achieved when $\Delta t < (\hat{\xi}_{k,optimal})^{-1} \Delta\tau$, thereby providing a lower bound on $\Delta\tau$. In the case of packet losses, the strategy employed to reduce the number of packet replacements is to maintain $\Delta\tau < \Delta t$. Thus, we derive the following inequality:

$$\hat{\xi}_{k,optimal} \Delta t < \Delta\tau < \Delta t \ , \tag{16}$$

(i.e., efficient merging capability in a timer interval n is achieved if $\Delta\tau/\Delta t$ is in the range $(\hat{\xi}_{k,optimal}, 1)$).

6.2. Second Scenario: Occurrence of Random Delays

We extend our model to take jitter into account. In our treatment, we consider the following sources of delay variability that may degrade the efficiency of merging procedures: (i) jitter experienced by the feedback

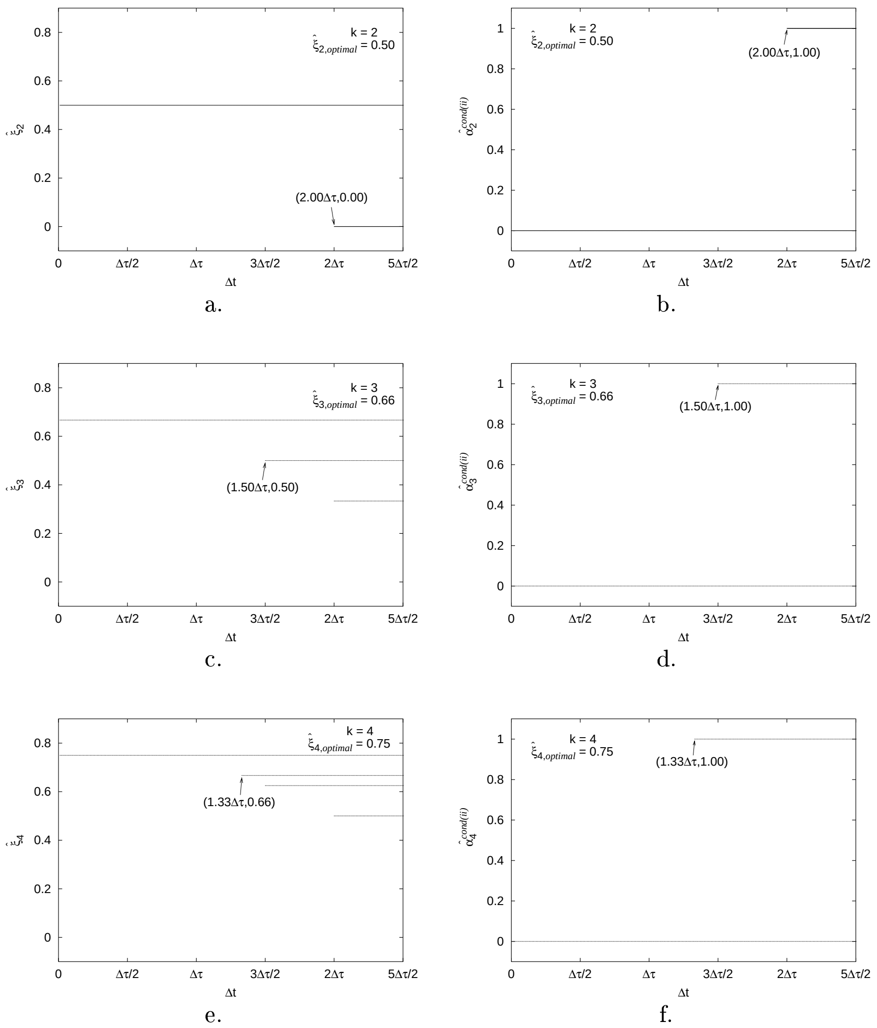

Figure 4. Theoretical results for the triggered scheme.

stream of each receiver; and (ii) variations in the merging time window size. Both tend to induce packet replacements.

Let δ be a positive bounded variable representing the variation in the merging time window size. Let l and m represent two receivers. Let J_l^{max} and J_m^{max} be, respectively, the maximum jitter experienced by the feedback stream of receivers l and m. We analyze a scenario in which a packet from receiver l arrives at the beginning of a timer interval n and a packet from receiver m arrives at the end of the same timer interval. As regards both the periodic and triggered schemes,

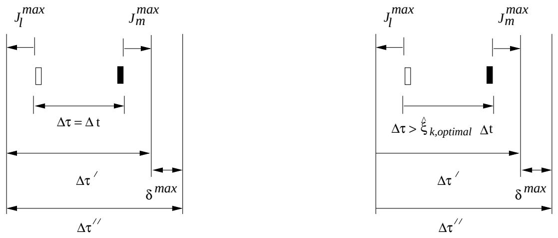

Figure 5. Worst case of delay variability for each timer scheme.

Figures 5.a. and 5.b., respectively, illustrate the worst case of delay variability that occurs in this scenario.

6.2.1. *Periodic Scheme*

In order for the merging scheme to work well, Δt must be sized sufficiently large with respect to jitter and window size variability. We approximate the cumulative jitter and variability as follows: to take jitter into account, the merger node should choose a merging time window size $\Delta\tau' = \Delta t + J_l^{max} + J_m^{max}$, as seen in Figure 5.a.. As the true window size can differ from the chosen value due to timer variability, the maximum possible window size is $\Delta\tau'' = \Delta\tau' + \delta^{max}$. We make no assumptions about which feedback source will be either receiver l or receiver m in each timer interval, so we make the conservative approximation $J_l^{max} + J_m^{max} = 2\max_{i=1,\ldots,k}[J_i^{max}]$. We wish for the influence of random delays to be an insignificant portion of $\Delta\tau''$. Thus Δt should be sized to be a large ratio of these delays. We define the ratio a as follows:

$$a = \frac{\Delta t}{\delta^{max} + 2\max_{i=1,\ldots,k}[J_i^{max}]}. \quad (17)$$

The performance is given in terms of packets replacements and merges attributable to *condition (i)*. In Section 7.2, we investigate the performance of merging procedures as a function of ratio a. It should be noted that ratio a is determined by the merger node and cannot be individually chosen by the receivers.

The occurrence of random delays does not change the relation between Δt and $\Delta\tau$. It does, however, introduce a lower bound on the feedback sending interval in order to achieve a desirable performance of merging procedures.

6.2.2. *Triggered Scheme*
From Figure 5.b., it follows that, due to jitter, the merger node should choose a merging time window size $\Delta\tau'$ such that $\Delta\tau' > \hat{\xi}_{k,optimal}\Delta t + J_l^{max} + J_m^{max}$. Due to timer variability, the maximum possible window size is $\Delta\tau'' = \Delta\tau' + \delta^{max}$. To reduce the probability of packet replacements, we should make $\Delta\tau'' < \Delta t$. Considering these conditions and making no assumptions about the feedback source that will be either receiver l or receiver m in each timer interval, it can be demonstrated that Δt should be sized to be a ratio of $k(\delta^{max} + 2\max_{i=1,\ldots,k}[J_i^{max}])$. Thus, we define the ratio d as follows:

$$d = \frac{\Delta t}{k(\delta^{max} + 2\max_{i=1,\ldots,k}[J_i^{max}])}. \tag{18}$$

Section 7.3 investigates the performance of merging procedures as a function of ratio d. The ratio d is also determined by the merger node.

The occurrence of varying delays changes the relation between Δt and $\Delta\tau$ and also introduces a lower bound on the feedback sending interval in order to achieve a desirable merging performance. The merging time window size we will adopt is $\Delta\tau = \hat{\xi}_{k,optimal}\Delta t + 2\max_{i=1,\ldots,k}[J_i^{max}]$.

7. Simulations

The results we present in this section were derived from simulations using the *ns* network simulator [2]. We adopt a 99% confidence level. It should be noted that in most cases confidence intervals shown in the figures are very small and can be seen as a single dot. In the next section, we describe the simulated scenarios. We present the results of the periodic and triggered schemes, respectively, in Section 7.2 and Section 7.3. We have chosen a uniform distribution rather than the more commonly assumed normal distribution for jitter in order to focus on the problem of high jitter variation (the uniform distribution poses a more challenging high variance than normal distribution).

7.1. Simulation Scenarios

The network topology considered is shown in Figure 1. Each link has 5 Mbps of bandwidth and a 2 ms delay. We have performed a set of simulations for $k = 10$ and $k = 100$. In each simulation, each receiver sends approximately 600 feedback packets. The start time of each receiver is uniformly chosen at random from the interval $[0,\ \Delta t)$. The merging window size variation δ of each timer interval is uniformly chosen at random from the interval $[0,\ 10]$ ms. After sending the first feedback packet, each receiver i sends its packets spaced apart by an interval uniformly chosen at random from the interval $[\Delta t - \frac{J_i^{max}}{2}, \Delta t + \frac{J_i^{max}}{2}]$. This procedure induces maximum jitter of J_i^{max} in the feedback stream of receiver i. In the simulations, J_i^{max} is identical for all receivers. We have performed simulations for J_i^{max} = [10 ms, 20 ms, 30 ms].

7.2. Periodic Scheme

Figure 6 shows the results for $k = 10$ and $k = 100$, with $\delta^{max} = 10\ ms$ and $J_i^{max} = 30\ ms$ for all $i \in [1, k]$. From Figure 6.a., we note that the fraction of merges due to *condition (i)* is greatly influenced by random delays when ratio a is low. This influence increases along with the number of receivers. The reason is as follows: the occurrence of random delays may induce the arrival of two packets from the same receiver within the same timer interval. If the second packet arrives immediately after a merging procedure that has been completed due to *condition (i)*, but before a timer expiration, a merging procedure due to *condition (ii)* will occur. If the second packet arrives before *condition (i)* holds, a packet replacement will take place. After any of these situations occurs within a timer interval, the subsequent timer interval may lack packets from the concerned receiver, which induces a merging procedure due to *condition (ii)*. Therefore, as k increases, there is a greater probability that both the number of merges attributable to *condition (ii)* and the number of packet replacements will also increase. Figure 6.b. confirms the effect of delay variability on the number of packet replacements as k increases. The Y-axis represents the percent of the total number of packets sent that have been replaced. We observe an average number of packet replacements equal to 3.6 when $k = 10$ and $a = 10$. As $k = 100$ and $a = 10$, the average number of packet replacements is 267. Increasing the value of a reduces the side effect of random delays.

Figure 6.c. shows the Global Filtering Level for $k = 10$ and $k = 100$. We note that the values are very close to the optimal Global Filtering Level. The more the ratio a is increased, the more the Global Filtering Level decreases towards $\xi_{k,optimal}$. For low values of a, the Global Filtering Level is less than $\xi_{k,optimal}$ because of the following: if, on the one hand, merges due to *condition (ii)* lead to a Global Filtering Level less than $\xi_{k,optimal}$, on the other hand, packet replacements lead to a Global Filtering Level greater than $\xi_{k,optimal}$. Thus, merges due to *condition (ii)* strongly impact the Global Filtering Level than do packet replacements.

Tables I and II summarize the outcomes of simulation studies with different J_i^{max} values. We note that the number of packet replacements

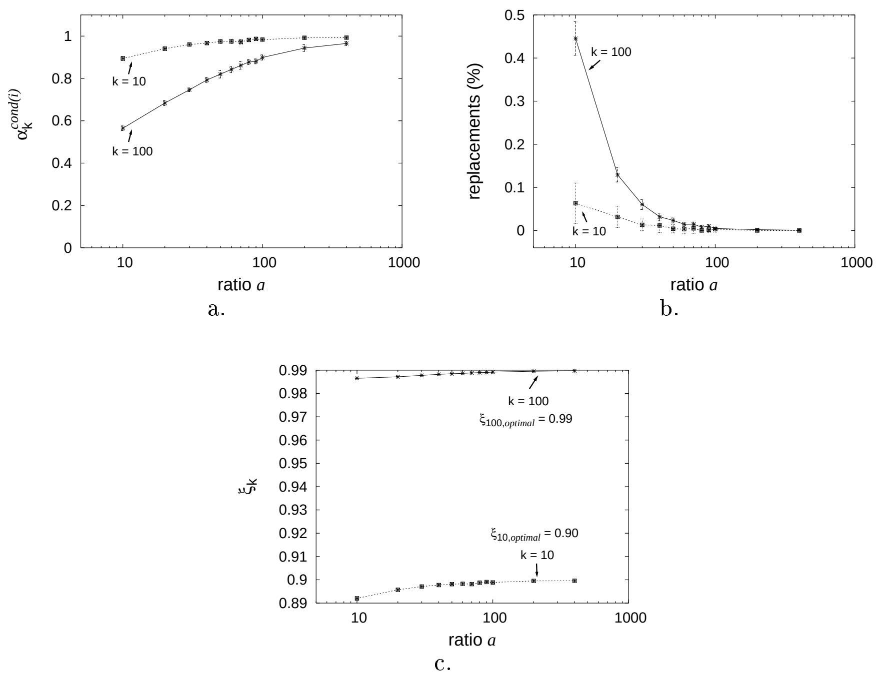

Figure 6. Periodic scheme ($\delta^{max} = 10\ ms$ and $J_i^{max} = 30\ ms$ for all $i \in [1, k]$): fraction of merges due to *condition (i)*; percentage of packet replacements; and the Global Filtering Level.

is low in all cases. In particular, we note that the number of merges due to *condition (i)* is greatly influenced by varying delays when $k = 100$. Increasing the ratio a reduces the side effect of varying delays. However, this implies an increase in the feedback sending interval. Assuming reasonably efficient merging, if a feedback latency on the order of a few seconds is needed to meet the application requirements, both varying delays and k should be low. In practice, this fact limits the applicability of the periodic scheme.

The minimum feedback sending interval that satisfies specified merging performance requirements can be determined by the merger node. To see this, consider the following: assume that $k = 10$, the maximum jitter observed is 10 ms, $\delta^{max} = 10\ ms$, and that the current feedback sending interval is 3 seconds. Under these conditions, $a = 100$. For this scenario, the expected performance of merging procedures in terms of merges due to *condition (i)* is around 98%. Now assume that the maximum jitter observed increases to 20 ms and that the performance of 98% must be maintained. From Table I, we verify that this performance requirement can be met if a is equal to 100, which implies a minimum

Table I. Summary of the periodic scheme results ($\delta^{max} = 10\ ms$).

		periodic scheme			
		J_i^{max} for all $i \in [1, k]$			
		10 ms		20 ms	
k	a	$\alpha_k^{cond(i)}$	$repl.(\%)$	$\alpha_k^{cond(i)}$	$repl.(\%)$
10	30	0.942 ± 0.004	0.005 ± 0.008	0.961 ± 0.007	0.010 ± 0.014
10	100	0.982 ± 0.001	0.000 ± 0.000	0.985 ± 0.008	0.000 ± 0.000
10	200	0.989 ± 0.004	0.000 ± 0.000	0.990 ± 0.006	0.000 ± 0.000
100	30	0.702 ± 0.007	0.042 ± 0.005	0.745 ± 0.008	0.054 ± 0.007
100	100	0.862 ± 0.005	0.004 ± 0.003	0.892 ± 0.008	0.005 ± 0.003
100	200	0.920 ± 0.010	0.001 ± 0.001	0.936 ± 0.007	0.001 ± 0.002

Table II. Summary of the periodic scheme results ($\delta^{max} = 10\ ms$).

		periodic scheme	
		J_i^{max} for all $i \in [1, k]$	
		30 ms	
k	a	$\alpha_k^{cond(i)}$	$repl.(\%)$
10	30	0.960 ± 0.007	0.013 ± 0.014
10	100	0.983 ± 0.006	0.003 ± 0.007
10	200	0.992 ± 0.006	0.000 ± 0.000
100	30	0.746 ± 0.009	0.060 ± 0.012
100	100	0.899 ± 0.012	0.005 ± 0.003
100	200	0.943 ± 0.016	0.002 ± 0.002

feedback sending interval of 5 seconds. In practice, a mechanism to inform receivers about the new feedback sending interval must be employed. One solution that does not require additional control packets is the following: a *feedback interval field* in feedback packets is used by the merger node to maintain its upstream neighbor node informed about the minimum feedback sending interval that is necessary to improve the performance of its merging procedures. When a merger node receives merged reports, the *feedback interval field* of outgoing packets must contain the maximum value between the feedback sending interval informed by its downstream neighbor nodes and the feedback sending interval that will improve the efficiency of its merging procedures. The incoming report at the source will carry the feedback sending interval value to be adopted by the receivers. The source informs the receivers of the new Δt by updating the *feedback interval field* in data packets.

7.3. Triggered Scheme

Figure 7 shows the results for $k = 10$ and $k = 100$, with $\delta^{max} = 10\ ms$ and $J_i^{max} = 30\ ms$ for all $i \in [1, k]$. We note that the triggered scheme performs very well even in the presence of varying delays. The behavior of the packet replacements is shown in Figure 7.a.. The Y-axis represents the percent of the total number of packets sent that have been replaced. We note an average value of 3.12 packet replacements when $k = 10$ and $d = 1$ ($\Delta t = 700\ ms$). When $k = 100$ and $d = 0.1$ ($\Delta t = 700\ ms$), we note the average number of 306 packet replacements. Thus, with a fixed feedback sending interval, the number of packet replacements increases as k increases. The reason is as follows: varying delays may induce the occurrence of more than one packet being received from the same receiver during the same timer interval. Thus, as k increases, the probability of packet replacements also increases. Aside from this fact, the triggered scheme still provides a low number of packet replacements in relation to the total number of packets sent.

Figure 7.b. shows the Global Filtering Level for $k = 10$ and $k = 100$. We see values that are very close to the optimal Global Filtering Level. The more the value of d increases, the more the Global Filtering Level decreases towards $\hat{\xi}_{k,optimal}$. For low values of d, the Global Filtering Level is greater than $\hat{\xi}_{k,optimal}$ for the following reason. Recall that merges due to *condition (ii)* lead to a Global Filtering Level less than $\hat{\xi}_{k,optimal}$ and that packet replacements lead to a Global Filtering Level greater than $\hat{\xi}_{k,optimal}$. From Figure 7.c. and Figure 7.d., we note that the number of merging procedures due to *condition (ii)* is approximately zero, which highlights the effect of packet replacements on the Global Filtering Level.

Tables III and IV summarize the results of simulation studies for different J_i^{max} values. These results show that the triggered scheme provides reasonably efficient merging even in the event of random delays.

As explained in Section 7.2, the minimum feedback sending interval that satisfies specified merging performance requirements can be determined by the merger node. We now provide an example using the triggered scheme. Assume that $k = 10$, the maximum jitter observed is 10 ms, $\delta^{max} = 10\ ms$, and that the current feedback sending interval is 300 ms. Hence, $d = 1$. Under this scenario, the expected performance of merging procedures in terms of merges due to *condition (i)* is maximized. On the other hand, the expected performance in terms of packet

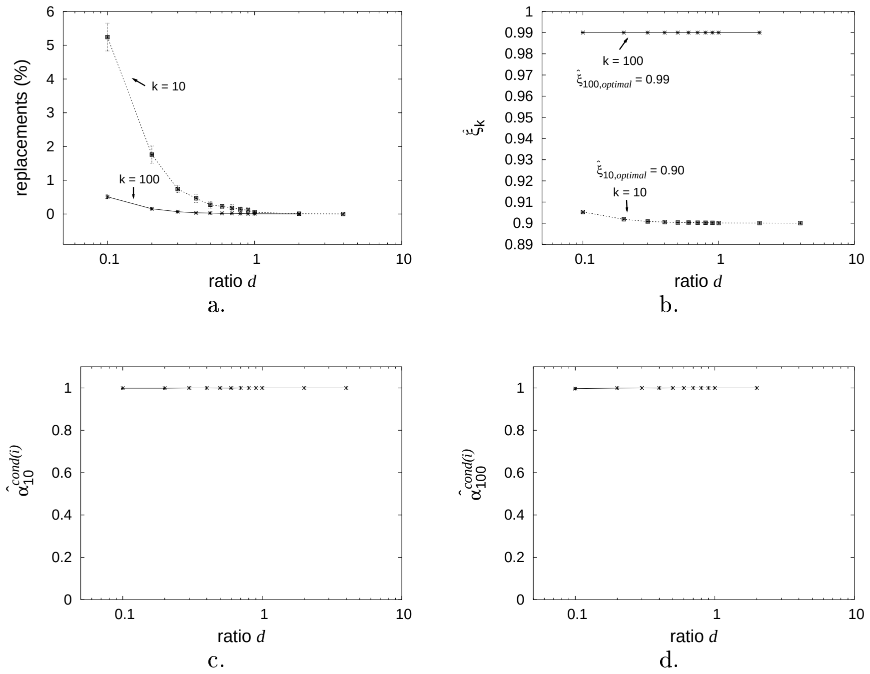

Figure 7. Triggered scheme ($\delta^{max} = 10\ ms$ and $J_i^{max} = 30\ ms$ for all $i \in [1, k]$): percentage of packet replacements; the Global Filtering Level; and fraction of merges due to *condition (i)* for $k = 10$ and $k = 100$.

Table III. Summary of the triggered scheme results ($\delta^{max} = 10\ ms$).

		triggered scheme			
		J_i^{max} for all $i \in [1, k]$			
		$10\ ms$		$20\ ms$	
k	d	$\hat{\alpha}_k^{cond(i)}$	$repl.(\%)$	$\hat{\alpha}_k^{cond(i)}$	$repl.(\%)$
10	0.1	0.998 ± 0.002	3.308 ± 0.336	0.997 ± 0.003	4.854 ± 0.216
10	0.3	0.999 ± 0.001	0.465 ± 0.094	0.999 ± 0.001	0.624 ± 0.129
10	1.0	1.000 ± 0.000	0.057 ± 0.033	0.999 ± 0.001	0.076 ± 0.040
10	2.0	1.000 ± 0.000	0.025 ± 0.022	1.000 ± 0.000	0.022 ± 0.019
100	0.1	0.998 ± 0.001	0.357 ± 0.024	0.998 ± 0.002	0.474 ± 0.046
100	0.3	0.999 ± 0.001	0.046 ± 0.012	0.999 ± 0.001	0.078 ± 0.019
100	1.0	0.999 ± 0.001	0.004 ± 0.003	1.000 ± 0.000	0.008 ± 0.002
100	2.0	1.000 ± 0.000	0.002 ± 0.002	1.000 ± 0.000	0.003 ± 0.005

Table IV. Summary of the triggered scheme results ($\delta^{max} = 10\ ms$).

		triggered scheme	
		J_i^{max} for all $i \in [1,k]$	
		$30\ ms$	
k	d	$\hat{\alpha}_k^{cond(i)}$	$repl.(\%)$
10	0.1	0.999 ± 0.009	5.243 ± 0.412
10	0.3	0.999 ± 0.001	0.741 ± 0.108
10	1.0	1.000 ± 0.000	0.052 ± 0.026
10	2.0	1.000 ± 0.000	0.010 ± 0.021
100	0.1	0.997 ± 0.004	0.510 ± 0.056
100	0.3	1.000 ± 0.000	0.069 ± 0.019
100	1.0	0.999 ± 0.001	0.008 ± 0.004
100	2.0	0.999 ± 0.001	0.004 ± 0.004

replacements is around 0.06%. Now assume that the maximum jitter observed increases to 30 ms and that the same performance must be maintained. From Table IV, we verify that this performance requirement can be met if d is equal to 1, which implies a minimum feedback sending interval of 700 ms.

8. Conclusion

In this paper, we have demonstrated how to increase the efficiency of a merger node for current source-adaptive layered multicast schemes. We have looked at two problems: feedback transmission and temporal merging. We have adopted an approach for transmitting feedback that, coupled with suitable timer settings at the merger node, improves merging efficiency. In our approach, each receiver regularly transmits consecutive feedback packets spaced apart by Δt.

We have studied two schemes for regulating the behavior of the merger timer: periodic and triggered. We have demonstrated that the triggered scheme outperforms the periodic scheme. This occurs because the triggered scheme has much better immunity to random delays (network jitter and variations in the merging time window size) than the periodic scheme. The suitable merging time window size to be used with the triggered scheme is $\Delta\tau = (1 - \frac{1}{k})\Delta t + 2\max_{i=1,\ldots,k}[J_i^{max}]$, where k is the number of receivers returning reports to the merger node, and J_i^{max} is the maximum jitter experienced by the feedback stream of receiver i. We have demonstrated that for the triggered scheme, Δt should be sized to be a ratio d of $k(\delta^{max} + 2\max_{i=1,\ldots,k}[J_i^{max}])$, where δ^{max} is the maximum variation in the merging time window size. We have determined through simulation studies suitable values of ratio d in order to achieve specified performance requirements. Assuming that $k = 100$, the maximum jitter observed is 10 ms, the maximum variability in the merging time window size is 10 ms, and that the current feedback sending interval is 300 ms ($d = 0.1$), we should expect the merging efficiency in terms of merges due to *condition (i)* to be maximized. The expected performance in terms of packet replacements is equal to 0.36%. In future works, we will extend our analytical model to determine ratio d values from specified performance requirements in terms of packet replacements and merges attributable to *condition (i)*.

We have also investigated the effects of temporal merging on feedback suppression. Our analysis has shown that there exists an optimal value for the Global Filtering Level. A Global Filtering Level that is greater than its optimal value increases efficiency for feedback implosion avoidance, but is inadequate for effective feedback.

The minimum feedback sending interval that meets specified performance requirements in terms of packet replacements and merges due to *condition (i)* is determined by the merger node and may change over time due to dynamic network conditions of delay variability and variations in the number of receivers returning reports. Thus, a mechanism to inform receivers about changes in the feedback sending interval must be employed. We have provided the following mechanism: a *feedback interval field* in feedback packets is used by the merger node to maintain its upstream neighbor node informed about the minimum feedback sending interval that is necessary to improve the performance of its merging procedures. When a merger node receives merged reports, the *feedback interval field* of outgoing packets must contain the maximum value between the feedback sending interval informed by its downstream neighbor nodes and the feedback sending interval that will improve the efficiency of its merging procedures. The incoming report at the source will carry the feedback sending interval value to be adopted by the receivers. The source informs the receivers of the new Δt by updating the *feedback interval field* in data packets.

The amount of generated feedback traffic grows linearly with the number of receivers. However, due to feedback aggregation, the amount of incoming feedback packets at the source is effectively equivalent to the amount of feedback generated by a single receiver. Moreover, it is desirable to limit the feedback traffic to a percentage of the session bandwidth. The feedback sending interval to accomplish this purpose is given by $T = \frac{number\ of\ participants\ .\ feedback\ packet\ size}{\%\ of\ the\ session\ bandwidth\ .\ session\ bandwidth}$. Since a merger node knows the number of its descendant receivers, this information can be carried by outgoing feedback packets and aggregated at upstream merger nodes to provide the source with a reasonably count of the number of receivers. The new Δt to be adopted is the maximum value between T and the feedback sending interval value carried by incoming reports at the source.

The mechanism for regulating the amount of generated feedback traffic does not require the use of additional control packets, thus reducing the overhead generated by control traffic presented in current proposals in the literature.

Acknowledgements

We are grateful to Timur Friedman, and the anonymous reviewers for their comments and suggestions that helped improve the quality of this paper. The first author also thanks CAPES/Brazil for the scholarship to support his Ph.D. studies.

References

1. Amorim, M. D., O. C. M. B. Duarte, and G. Pujolle: 1999, 'Single-loop Packet Merging for Receiver Oriented Multicast Multi-Layered Video'. In: *Proc. of International Conference in Computer Communications*. Tokyo, Japan.
2. Bajaj, S. et al.: 1999, 'Improving Simulation for Network Research'. Technical report, University of Southern California, Department of Computer Science.
3. Blake, S. et al.: 1998, 'An Architecture for Differentiated Services'. Internet RFC 2475.
4. Costa, L. H. M. K., S. Fdida, and O. C. M. B. Duarte: 2001, 'A Scalable Algorithm for Link-state QoS-based Routing with Three Metrics'. In: *Proc. of IEEE ICC'2001*. Helsink, Filand.
5. Crawley, E., R. Nair, B. Rajagopalan, and H. Sandick: 1998, 'A Framework for QoS-Based Routing'. Internet RFC 2386.
6. da Silva Gonçalves, P. A., J. F. de Rezende, and O. C. M. B. Duarte: 2000, 'An Active Service for Multicast Video Distribution'. *Journal of the Brazilian Computer Society* **7**(1), 43–51.
7. de Amorim, M. D., O. C. M. B. Duarte, and G. Pujolle: 2000, 'Multi-criteria arguments for improving the fairness of layered multicast applications'. *Lecture Notes in Computer Science* (1815), 1–10.
8. de Amorim, M. D., O. C. M. B. Duarte, and G. Pujolle: 2001, 'Application-Aware Multicast'. In: *Proc. of GLOBECOM'2001*. San Antonio, Texas.
9. Diot, C., C. Huitema, and T. Turletti: 1995, 'Multimedia applications should be adaptive'. In: *Proc. of HPCS'95*.
10. Kasera, S. K. et al.: 2000, 'Scalable Fair Reliable Multicast Using Active Services'. *IEEE Network* **14**(1), 48–57.
11. Li, X., S. Paul, and M. H. Ammar: 1998, 'Layered Video Multicast with Retransmission (LVMR): Evaluation of Hierarchical Rate Control'. In: *Proc. of IEEE INFOCOM'98*. San Francisco, CA, pp. 1062–1072.

12. Ma, Q. and P. Steenkiste: 1997, 'Quality of Service Routing for Traffic with Performance Guarantees'. In: *Proc. of IFIP International Workshop on Quality of Service*. New York, USA.
13. McCanne, S., V. Jacobson, and M. Vetterli: 1996, 'Receiver-driven Layered Multicast'. In: *Proc. of SIGCOMM'96*. Stanford, CA, pp. 117–130.
14. Papadopoulos, C., G. Parulkar, and G. Varghese: 1998, 'An Error Control Scheme for Large-Scale Multicast Applications'. In: *Proc. of IEEE INFOCOM'98*. San Francisco, CA, pp. 1188–1196.
15. Rejaie, R., D. Estrin, and M. Handley: 1999, 'Quality Adaptation for Congestion Controlled Video Playback over the Internet'. In: *Proc. of ACM SIGCOMM'99*. Cambridge, MA.
16. Rejaie, R., M. Handley, and D. Estrin: 2000, 'Layered Quality Adaptation for Internet Video Streaming'. *IEEE Journal on Selected Areas in Communications* **18**(12).
17. Shin, J., J. Kim, and C.-C. J. Kuo: 2001, 'Quality of Service Mapping Mechanism for Packet Video in Differentiated Services Network'. *IEEE Transactions on Multimedia* **3**(2), 219–231.
18. Vickers, B. J., C. Albuquerque, and T. Suda: 1998, 'Adaptive Multicast of Multi-Layered Video: Rate-Based and Credit-Based Approachs'. In: *Proc. of IEEE INFOCOM'98*. San Francisco, CA.
19. Vickers, B. J., M. Lee, and T. Suda: 1997, 'Feedback Control Mechanisms for Real-Time Multipoint Video Services'. *IEEE Journal on Selected Areas in Communications* **15**(3), 512–530.
20. Wang, Z. and J. Crowcroft: 1996, 'Quality of Service Routing for Supporting Multimedia Applications'. *IEEE Journal on Selected Areas in Communications* **14**(7).
21. Whetten, B. and G. Taskale: 2000, 'An Overview of Reliable Multicast Transport Protocol II'. *IEEE Network* **14**(1), 37–47.
22. Ziviani, A., J. F. de Rezende, O. C. M. B. Duarte, and S. Fdida: 2001, 'Evaluating Voice Traffic in a Differentiated Services Environment'. In: *Proc. of 17th International Teletraffic Congress - ITC17*. Salvador, Brazil.

Address for Offprints:
Paulo Gonçalves - LIP6 - UPMC
8, rue du Capitaine Scott
75015 - Paris - France
e-mail: Paulo.Goncalves@lip6.fr
Phone: +33 1 44 27 61 88
Fax: +33 1 44 27 74 95