

# On the Scaling of Feedback Algorithms for Very Large Multicast Groups

Thomas T. Fuhrmann <sup>a,1</sup> Jörg Widmer <sup>b</sup>

<sup>a</sup>*The Boston Consulting Group, Munich, Germany*

<sup>b</sup>*Praktische Informatik IV, University of Mannheim, Germany*

---

## Abstract

Feedback from multicast group members is vital for many multicast protocols. In order to avoid feedback implosion in very large groups feedback algorithms with well behaved scaling-properties must be chosen. In this paper we analyse the performance of three typical feedback algorithms described in the literature. Apart from the basic trade-off between feedback latency and response duplicates we especially focus on the algorithms' sensitivity to the quality of the group size estimation. Based on this analysis we propose a generalized framework for feedback algorithms and especially give recommendations for the choice of well behaved feedback mechanisms that are suitable for very large groups.

*Key words:* multicast feedback; feedback suppression; scalability

---

## 1 Introduction

Most multicast protocols require some kind of feedback from the members of the multicast group: IGMP messages with multicast routing protocols, negative acknowledgements to initiate retransmissions with reliable multicast, and many other kinds of responses with various other kinds of protocols.

In traditional networks, these mechanisms rarely pose a big problem: for example, IGMP is used on a per-router basis with only a few hosts connected to the routers' physical links. Even if all these hosts simultaneously answer a feedback query for which one response message would have been sufficient not much harm is done with respect to the network load because the number of hosts directly connected to a router is small.

---

<sup>1</sup> done while with Praktische Informatik IV, University of Mannheim, Germany

In very large broadcast networks, e.g. satellite networks, one might expect  $10^6$  or even more hosts to be connected to the same physical link. Here feedback algorithms must carefully avoid feedback implosion, i.e. a large avalanche of identical responses to a query that could be answered by any single one of the hosts. Even more, in satellite networks round-trip times for responses are rather large. That means, if one of the hosts answers a query other hosts will not immediately notice that a response was already given. Hence, even more feedback duplicates will be sent to the network.

In this work we analyse these problems by studying three prototypical feedback algorithms with respect to their feedback latency and the expected number of responses for each query. In section 2 we present a mathematical analysis of the feedback latencies and the number of response messages of the three algorithms. Section 3 gives a comparison of these properties in the limit for large groups. Based on these findings, a general form of feedback mechanism is developed in section 4. The three discussed algorithms can be expressed as specific instances of the general form. In section 5, these analytical results are compared with results from simulations. Section 6 relates our work to other results obtained recently in this area of study. Section 7 draws conclusions from our studies and recommends the *exponential feedback raise* algorithm as the algorithm of choice for very large networks.

## 2 Analysis of three feedback algorithms

In this section, we examine three feedback algorithms that are prototypical for algorithms currently being deployed or recently proposed. They all address the *at-least-one* scenario in which a single response to a request suffices but multiple identical responses from different group members will do no harm except for the superfluous network load. We assume that the request is multicast to the whole group, whereas the corresponding responses are unicast to the sender. In order to allow for the suppression of further responses, the sender confirms the reception by multicasting a confirmation back to the group.

Compared to the case where the responses themselves are multicast, this scenario causes less traffic but increases the network-latency.<sup>2</sup> Furthermore, it can also be applied in single-source multicast networks where only the sender but not the receivers can multicast packets. Multicasting or unicasting feedback does not affect the general suppression characteristics of the investigated feedback mechanisms. For the mathematical analysis we additionally assume the latency to be constant for all group members. The question of packet-loss

---

<sup>2</sup> Note that for e.g. unidirectional satellite links [5,8] the network latency is *not* increased since all traffic is passed through a central network node.

and heterogeneous network latencies will be dealt with in the following sections. It will be shown that in the latter case the feedback latency is further reduced.

Although this simplified scenario is independent from the actual network-topology, a typical example for our scenario is a satellite serving a large number of small networks, e.g. home-networks, with multicast-traffic [5,8]. Since here a single link serves a large number of hosts, the feedback algorithms cannot rely on inner network nodes for feedback-suppression. Accordingly, all of the three algorithms studied here can be implemented as pure end-to-end protocols.

### 2.1 Equally distributed feedback

The classical algorithm for feedback suppression with random timers uses equally distributed response probabilities. A typical implementation of this algorithm is SRM [6]. The algorithm<sup>3</sup> can be described as follows:

**Algorithm 1** *Let  $T$  be a constant upper time limit. Upon reception of a feedback request sample  $x \in [0, 1)$  from a uniform distribution and start a timer  $t$ . If a feedback response is confirmed before  $t \geq xT$  holds, the clock is stopped and no feedback response is sent. Otherwise, a response is sent as soon as the given condition is satisfied.*

Let  $n$  be the number of potential responders. We begin our analysis by noting that  $(1 - x)^n$  is the probability that all  $x_i$  with  $i = 1 \dots n$  are larger than  $x$ . Hence the probability that  $x_{min} = \min\{x_1, \dots, x_n\} \in [x, x + dx]$  is  $n(1 - x)^{n-1}dx$ . The time corresponding to that choice of  $x$  is  $t = xT$ . Hence we obtain as expected value of the feedback latency  $L$

$$E[L] = T \int_0^1 n(1 - x)^{n-1}x dx = \frac{T}{n + 1} \quad (1)$$

Since the feedback-responses are distributed equally over the interval  $T$  the expected value for the number  $R$  of responses is given by

$$E[R] = n \frac{\tau}{T} \quad (2)$$

where  $\tau$  is the network's latency.

---

<sup>3</sup> We do not consider the adaption of the answer interval to the group size and network-latency here. This topic will be discussed in the following sections.

## 2.2 Independent feedback rounds

Let us now study an algorithm that makes use of the concept of feedback rounds [17]. It can be phrased as follows:

**Algorithm 2** *Let  $\tau$  be the network's latency and  $p \in (0, 1]$  a constant. Upon reception of a feedback request sample  $x \in [0, 1)$  from a uniform distribution, start a timer  $t$ , and immediately send a response if and only if  $x < p$ . If after the time  $t = \tau$  no response has been confirmed start a new round i.e. act as if another feedback request was received.*

Again, let  $n$  be the number of hosts that can send a response. The number of feedback rounds can easily be calculated as  $1 + (1 - p)^n + (1 - p)^{2n} + \dots = \frac{1}{1 - (1 - p)^n}$ . Hence the number of additional rounds after the first round is  $\frac{1}{1 - (1 - p)^n} - 1 = \frac{(1 - p)^n}{1 - (1 - p)^n}$ . From this we immediately obtain the expected value for the feedback latency:

$$E[L] = \tau \frac{(1 - p)^n}{1 - (1 - p)^n} \quad (3)$$

The algorithm does not guarantee the reception of a feedback message within a certain time limit  $T$ .

The expected value for the number of feedback responses is given by the product of the number of rounds and the expected value for each round. The latter is given by  $np$ . Hence we obtain:

$$E[R] = \frac{np}{1 - (1 - p)^n} \quad (4)$$

The latter might strike since one might have expected  $np$  as result arguing that only the last round contributes to the number of responses and thus the number of rounds must not be taken into account. Following this approach, one would however have to use the expected value of responses under the condition that at least one response is sent. Both approaches arrive at the same result.

A contour plot of the two expected values is shown in Figure 1. In order to simplify the comparison with other feedback-algorithms the response probability  $p$  has been expressed as  $p = 1/N$ .

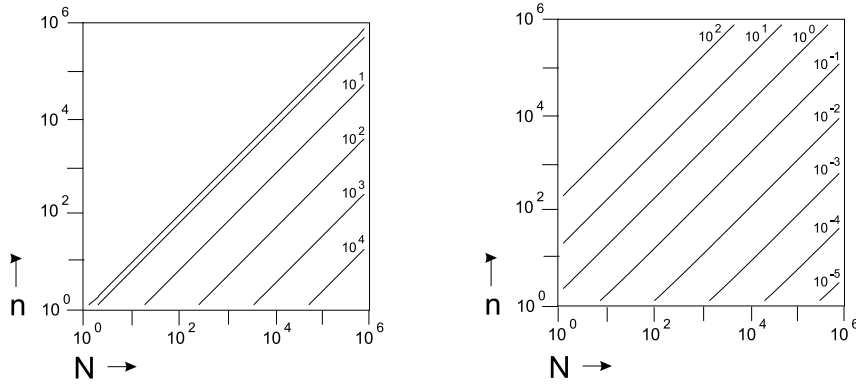


Fig. 1. Feedback latency (left) and feedback responses (right) for independent feedback rounds (double-logarithmical plot). Contour lines *logarithmically* indicate values from  $10^{-1}\tau$  to  $10^4\tau$  and surplus responses from  $10^2$  to  $10^{-5}$ .

### 2.3 Exponential feedback raise

A third alternative originally proposed by Bolot, Turetti, and Wakeman [1] and in an improved version extensively studied by Nonnenmacher and Biersack [13] is the exponential adaption of the feedback probability. For our purposes we use their mechanism in the following formulation:

**Algorithm 3** *Let  $N$  be the estimated number of group members and  $T$  be a constant upper time limit. Upon reception of a feedback request sample  $x \in [0, 1)$  from a uniform distribution and start a timer  $t$ . If a feedback response is confirmed before  $x < N^{t/T-1}$  holds, the clock is stopped and no feedback response is sent. Otherwise a response is sent as soon as the given condition is satisfied.*

Let us now analyse this algorithm: Like in the first algorithm studied  $T$  is the guaranteed upper limit to the time when the first response is sent. To derive the expected values we use the probability distribution for the least value of  $x$  chosen by the group of potential responders. As derived above the probability that  $x_{min} = \min\{x_1, \dots, x_n\} \in [x, x + dx]$  is  $n(1-x)^{n-1}dx$ . The time corresponding to that choice of  $x$  is  $t = T \cdot (1 + \log_N x)$ .

With these two results we obtain the feedback latency, i.e. the expected value for the earliest response:

$$E[L] = T \int_{N^{-1}}^1 n(1-x)^{n-1}(1 + \log_N x) dx \quad (5)$$

$$= \frac{T}{\ln N} \int_{1/N}^1 \frac{(1-x)^n}{x} dx \quad (6)$$

Similarly, we can calculate the expected value for the number of responses. Assume that  $x$  is the smallest value chosen in the group. If  $x \leq N^{-1}$  a response will be immediately sent. However, due to the network's latency  $\tau$  duplicate responses will be received from all members that chose their value  $x_i$  in the interval  $[x, N^{\tau/T-1})$ .

If  $x > N^{-1}$  the earliest response will be sent at a time  $t > 0$ . Duplicate responses will then be received from all members that chose their value  $x_i$  in the interval  $[N^{t/T-1}, N^{\frac{t+\tau}{T}-1})$ . Using  $x = N^{t/T-1}$  this interval can be written as  $[x, xN^{\tau/T})$ .

Under the condition that all responses after the first response are distributed equally in the interval  $[x, 1)$  we find the following expected values for duplicate responses in these two cases

$$(n-1) \frac{N^{\tau/T-1} - x}{1-x}$$

and

$$(n-1) \frac{xN^{\tau/T} - x}{1-x}$$

If the earliest response is sent after  $t = T - \tau$  no suppression can take place any more. Clearly, the probability for this case is  $(1 - N^{-\tau/T})^n$ . Altogether we find

$$E[R] = \int_0^{1/N} (n-1) \frac{N^{\tau/T-1} - x}{1-x} \cdot n(1-x)^{n-1} dx \quad (7)$$

$$+ \int_{1/N}^{N^{-\tau/T}} (n-1)x \frac{N^{\tau/T} - 1}{1-x} \cdot n(1-x)^{n-1} dx \quad (8)$$

$$+ (n-1)(1 - N^{-\tau/T})^n \quad (9)$$

$$= N^{\tau/T} \left( \frac{n}{N} + \left(1 - \frac{1}{N}\right)^n - \left(1 - \frac{1}{N^{\tau/T}}\right)^n \right) \quad (10)$$

A plot of these two expected values is shown in Figure 2.

### 3 Comparison of the algorithms for very large groups

Based on the results derived above we can now compare the suitability of these three feedback algorithms in the context of very large networks. Concrete values for given  $n$  and  $N$  up to  $10^6$  can be read off from Figures 1 and 2. For

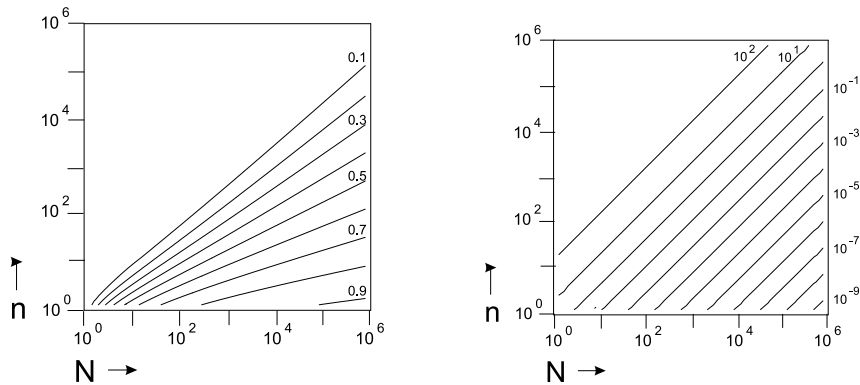


Fig. 2. Feedback latency (left) and feedback responses (right) for Exponential Feedback Raise (double-logarithmic plot). Contour lines *linearly* indicate values from  $0.1T$  to  $0.9T$  and *logarithmically* indicate surplus responses from  $10^2$  to  $10^{-9}$ .

a more thorough judgment of the algorithms' behaviour in the limit of very large groups we will further analyse the formulae derived above.

As one might expect, we need some estimation of the group size  $N$  in order to optimise the algorithms' parameters. However, good group size estimations cannot always be given. Consider for example a reliable multicast transmission. Depending on where the packet loss occurs the number of hosts that need to send a negative acknowledgement (NAK) can vary greatly from packet to packet. If the loss occurs near the multicast sender almost all receivers might be potential NAK-senders. Otherwise, only a few hosts need to send a NAK.

Owing to this fact it is important to design a feedback algorithm that is insensitive to large variations of the group size. However, concerning gross underestimation of the group size a fixed limit can be given that no algorithm studied here (i.e. an algorithm based on independent identical hosts in a network with non-vanishing latency) can overcome:

**Lemma** *Underestimation of the group size results in an asymptotically linear increase of the number of feedback responses.*

*Proof:* Let  $P(t)$  be the probability of answering before time  $t$ . Without loss of generality we can assume that  $P(t) > 0$  for  $t > 0$ . Then  $P(\tau) > 0$  is the finite probability for a host to answer before the suppression mechanism can be effective. Hence  $E[R] \geq nP(\tau)$ . On the other hand  $\forall \epsilon > 0 : \lim_{n \rightarrow \infty} nP(\epsilon) > 1$ . Thus for  $n \rightarrow \infty$  suppression immediately sets in at  $t = \tau$  and we have  $E[R] = nP(\tau)$ .

In order to avoid this linear behaviour we will want to operate our system such that our estimation is an upper limit for the group size. Unlike the actual group size such a limit can usually be estimated rather easily (e.g. one can assume that the total number of installed hosts is known to the network provider).

Due to this characteristic it is hence crucial for the feedback algorithms to be insensitive to overestimations of the group size.

### 3.1 Equally distributed feedback

The equally distributed feedback algorithm shows a rather simple behaviour in the limit for large groups. Since  $\lim_{n \rightarrow \infty} E[R] = \infty$  for fixed  $T$  we have to adapt  $T$  to the group size. Generally, by eliminating  $T$  we find a trade-off between feedback-latency  $E[L]$  and response duplicates  $E[R]$ :

$$\lim_{n \rightarrow \infty} E[L] E[R] = \tau \quad (11)$$

Although we expect a feedback latency in the order of the network's latency and a number of response duplicates of order one, the accuracy of our estimation  $N$  of the group size is crucial. Overestimation or underestimation of  $N$  linearly affects both  $E[L]$  and  $E[R]$ . Thus this algorithm is not well suited for very large networks.

### 3.2 Independent feedback rounds

As mentioned above, collecting responses for immediate sending at the beginning of each round can reduce the feedback latency by up to  $\tau$  while it ideally preserves the number of responses. However, rather than using this simple method we investigated a slightly different algorithm that uses *independent* rounds.

Let us now analyse this algorithm in the limit of large groups: Setting  $p = \frac{1}{N}$  and  $n = \alpha N$  we have in the limit  $N \rightarrow \infty$

$$E[L_\infty] = \tau \frac{e^{-\alpha}}{1 - e^{-\alpha}} = \frac{\tau}{e^\alpha - 1} \quad (12)$$

$$E[R_\infty] = \frac{\alpha}{1 - e^{-\alpha}} \quad (13)$$

As expected we see that if we underestimate the size of the group ( $\alpha \gg 1$ ) the number of feedback responses grows asymptotically linearly while the feedback latency vanishes. However, if we overestimate the size of the group ( $\alpha \ll 1$ ) we see that the feedback latency grows according to  $E[L] \simeq \frac{\tau}{\alpha + \dots}$  while the number of feedback responses  $E[R] \rightarrow 1$ .



This is once more an undesired behaviour since estimating the group size is hence again as decisive as the right choice of a passage between Scylla and Charybdis. As with the equally distributed feedback no choice of parameters can guarantee a controllable behaviour for all  $\alpha \in [0, 1]$ . Even worse, due to the independence of the feedback rounds we have  $E[L] E[R] \simeq \frac{\tau}{\alpha}$  for  $\alpha \ll 1$ . Thus, this algorithm is as uncommendable as the previous one for the situation under investigation.

### 3.3 Exponential feedback raise

As above we set  $n = \alpha N$  for our analysis of the  $N \rightarrow \infty$  limit. Additionally, we choose  $T$  such that  $N^{\tau/T} = e^\beta = \text{const}$ , i.e. we set

$$\beta = \frac{\tau}{T} \ln N \quad (14)$$

With this adaption we now have

$$E[R_\infty] = (\alpha + e^{-\alpha})e^\beta \quad (15)$$

From the construction of the algorithm we know that  $T$  is an upper limit for the feedback latency. Setting  $n = 1$  in (5) we find accordingly

$$E[L_\infty] \leq T - T \left( \frac{N-1}{N \ln N} \right) \approx T - \frac{\tau}{\beta} \quad (16)$$

Noting that for large  $n$  the main contribution to the integral comes from the lower boundary of the integral, we can make the following estimation for an upper limit in the  $\alpha \rightarrow 1$  case:

$$E[L_\infty] \leq \frac{T}{\ln N} \int_{1/N}^1 \frac{(1-x)^n}{1/N} dx \quad (17)$$

$$= \frac{T}{\ln N} \frac{N}{n+1} \left(1 - \frac{1}{N}\right)^{n+1} \approx \frac{\tau e^{-\alpha}}{\alpha \beta} \quad (18)$$

Numerical integration for  $N \simeq 10^5$  shows that  $\lim_{\alpha \rightarrow 1} E[L_\infty] \simeq 0.22 \frac{\tau}{\beta}$  which furthermore improves our gauge. Altogether we can thus say that:

$$\frac{T}{10 \log_{10} N} \leq E[L_\infty] \leq T \quad (19)$$

Hence, unlike in the two previous algorithms both expected values remain rather insensitive to variations of  $\alpha$ . Even in the limit  $\alpha \rightarrow 0$  both expressions remain finite. This is a strong indication that this mechanism is well suited for very large networks.

According to (15) the number of duplicates changes only by a factor of 1.3679 between the two extreme cases  $n = 1$  and  $n = N$ . Additionally, even in the worst case scenario the feedback latency remains below the threshold of  $T$ . From (14) we read that the choice of this threshold also determines the number of expected feedback duplicates. If we denote the expected value of responses in the  $\alpha \rightarrow 0$  limit by  $E[R_0]$  we have the following relation:

$$T = \tau \log_{E[R_0]} N \quad (20)$$

## 4 A generalized feedback algorithm

Based on the insights gained so far we will now discuss a generalized feedback algorithm that unifies the properties of the three algorithms.

### 4.1 Feedback rounds versus continuous feedback raise

Comparing the algorithms previously studied we see that sending a response at time  $t$  with  $0 < t < \tau$  is suboptimal since no suppression can take place before  $t = \tau$  but the latency is increased compared to an immediate response at  $t = 0$ . The second algorithm respects this fact by sending feedback in rounds, i.e. it sends a certain number of responses only at the beginning of a feedback interval of length  $\tau$ . No further responses are sent before the beginning of an eventual next round.

With the help of this insight both of the other algorithms can be improved as well. A straight-forward general implementation of this mechanism can be phrased as follows:

**Algorithm for feedback rounds** *Divide the interval  $[0, T]$  into sub-intervals  $[k\tau, (k+1)\tau]$  where  $k \in \{0, \dots, \lfloor \frac{T}{\tau} \rfloor\}$ . Send all responses that would be sent in a sub-interval at the beginning of that sub-interval.*

As we will see, this general mechanism reduces the feedback latency while it ideally preserves the number of response duplicates. A detailed analysis of a specific version of exponential feedback raise with rounds can be found in Bolot et al. [1]. Here, we will therefore focus on more principal aspects. As depicted in Figure 3, the general mechanism for the introduction of feedback

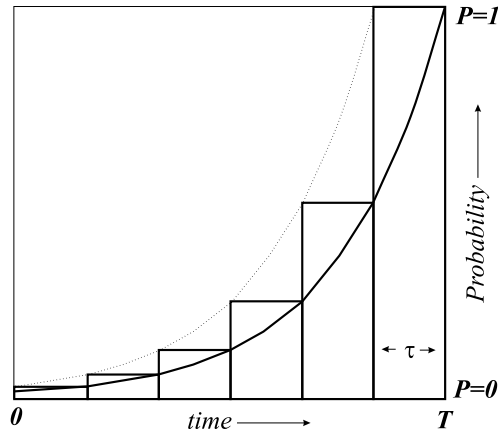


Fig. 3. Exponential feedback with and without rounds.

rounds raises the response probability for each host at  $t = k\tau$  to  $P(t + \tau)$ . That means that the hosts' response probability is given by a step-function  $P_s(t)$  that lies between the original function  $P_0(t)$  and  $P_1(t) = P_0(t + \tau)$ .

Since generally

$$E[L] = \int_0^T t \cdot n(1 - P(t))^{n-1} P'(t) dt \quad (21)$$

$$= \int_0^T (1 - P(t))^n dt \quad (22)$$

the feedback latency is given by the area above the function  $\hat{P}(t)$  where  $\hat{P}(t) = 1 - (1 - P(t))^n$ . Noting that  $P_1$  results in a feedback latency that is reduced by  $\tau$  as compared to  $P_0$  the feedback latency is generally reduced by  $\Delta L$  with  $0 < \Delta L < \tau$ . For  $T \gg \tau$  the area between the functions  $\hat{P}_0$  and  $\hat{P}_s$  approximates the area between the functions  $\hat{P}_1$  and  $\hat{P}_s$  and we have  $\Delta L \simeq \frac{\tau}{2}$ .

However, if we slightly underestimate the network latency  $\tau$  the hosts will perform an additional superfluous feedback round. This will result in an increase of feedback responses by a factor of  $N^{\tau/T}$ . In networks with heterogeneous latencies we must hence either choose  $\tau$  to be the maximal latency or a subgroup of hosts will perform an additional superfluous feedback round. Since on the other hand overestimation of the network latency leads to a linearly increased feedback latency,  $\tau$  should not be chosen too generously. Resolving this trade-off thus requires a rather good knowledge of the network latencies which is not necessary in the case of continuous feedback.

## 4.2 A generalized feedback algorithm

Having formulated a general mechanism to transform a continuous feedback algorithm into a round-based feedback algorithm, we can now more deeply analyse the relationship between the algorithms studied above. In order to do so, we transform all three algorithms into the same representation. The specific form of the algorithm is given by a distribution function  $f(t)$  which determines the feedback behaviour.

**General feedback algorithm** *Upon reception of a feedback request sample  $x \in [0, 1)$  from a uniform distribution. Send a feedback response as soon as  $x \leq f(t)$  where  $t$  denotes the time after the reception of the request. If however another group member's response has been confirmed before that time no response should be sent.*

As stated above, the distribution function  $f$  is decisive for the feedback behaviour. For the first algorithm given in section 2 we read off  $f(t) = t/T$  and similarly  $f(t) = N^{t/T-1}$  for the third. For the second algorithm  $f$  is determined by the probability for a group member to respond before or in the  $k^{\text{th}}$  round. In the first round the probability to respond is  $p$ . Hence  $f(t) = p$  for  $0 < t < \tau$ . In order to calculate  $f$  for the further rounds we note that  $(1 - p)^{k+1}$  is the probability that no response is sent before round  $k + 1$ . Hence the probability that the first response is sent before or in the  $k^{\text{th}}$  round is  $1 - (1 - p)^{k+1}$ , i.e.  $f(t) = 1 - (1 - p)^{k+1}$  for  $k\tau < t < (k + 1)\tau$ .

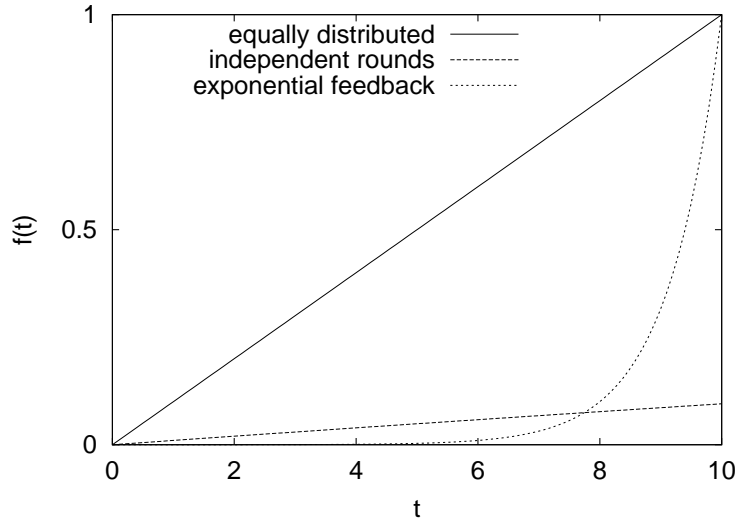
In order to simplify the analysis we will now drop the notion of rounds and consider the continuous distribution function  $f(t) = 1 - (1 - p)^{t/\tau+1}$ . As already stated, the error that is hereby introduced slightly increases the number of response duplicates while reducing feedback latency. A round-based algorithm can always be recovered by application of the mechanism described above.

Let  $p(t)dt$  be the probability for a host to respond in the time interval  $[t, t + dt]$  under the condition that no response was confirmed before  $t$ . Then the probability  $f(t)$  that a response is confirmed by  $t$  is determined by the following differential equation:

$$f'(t + \tau) = p(t)(1 - f(t)) \quad (23)$$

An overview of the three algorithms is given in figure 4.

Assuming that we know the actual size of the group we can always choose  $p(0)$  such that with any desired probability at least one response is sent immediately after the request was received. By this, the feedback latency is minimized. Only in the unlikely case that no response is sent at  $t = 0$  the feedback process



Equally distributed feedback	$f(t) = t/T$	$p(t) = \frac{1}{T-t}$
Independent feedback rounds	$f(t) = 1 - (1 - \frac{1}{N})^{t/\tau+1}$	$p(t) = \frac{1}{\tau} \frac{N-1}{N} \ln \frac{N}{N-1}$
Exponential feedback raise	$f(t) = N^{t/T-1}$	$p(t) = \frac{N^{\tau/T} \ln N}{T} \frac{1}{N^{1-t/T}-1}$

Fig. 4. Distribution functions for the three algorithms with  $N = 100000$ ,  $T = 10$ , and  $\tau = 0.001$  (Note that the ratio of  $\tau$  to  $T$  is unrealistic for most real-world scenarios and has only been chosen to better visualize independent feedback rounds.)

continues. If we are sure about the group size and no overall time limit is given, there is no reason to modify the response probability over time. Hence  $p(t) = const$ , i.e. the second algorithm is optimal.

If a time limit  $T$  is given, we have to choose the response probability such that  $\lim_{t \rightarrow T} p(t) \rightarrow \infty$ . This is guaranteed by the first and the third algorithm. The latter furthermore takes into account, that not receiving response confirmations can also be caused by an overestimation of the group size and accordingly adjusts  $p$ .

Before concluding our analysis, we shortly address the effect of packet loss. Since all algorithms discussed here are based on independently acting hosts, a lost response packet does not harm the principal effectiveness of the feedback mechanism. If the response is lost before it is confirmed by the sender, a loss rate of  $p$  merely reduces the effective group size from  $n$  to  $(1-p)n$ . If on the other hand a fraction  $q$  of the group receives the feedback, the effective group size is further reduced to  $(1-p)(1-q)n$ . Due to the algorithms' insensitivity to gross variations in the group size, packet loss only marginally affects the results given above.

## 5 Simulation results

To investigate the applicability of the different feedback mechanisms discussed in section 2, a simulation model of the algorithms was studied. For an upper limit of  $N = 10^6$  hosts we examined groups with  $n = 1$  to  $n = 10^6$  actual hosts. To be able to abstract from a specific network topology and independent receiver-to-receiver delays, we considered the case of unicast feedback to the sender as discussed in section 2. Note that this increases the feedback delay and the number of responses and thus represents an upper bound for the general scenario. All results were averaged over 1000 simulation runs to minimize the impact of statistical errors.

The feedback mechanism with independent feedback rounds and constant feedback probability has optimal characteristics when the number of participants is known (i.e.  $p = 1/n$ ). As shown in Figure 5, the average number of feedback responses is lower than that of all other mechanisms without impairing the feedback latency. However, in most cases the group size is unknown to the sender (or the feedback mechanism itself is used to estimate the group size [12]) and an estimate  $N$  has to be used. As discussed in section 2.2, the feedback latency increases proportionally to the ratio of  $N$  to  $n$  which makes the mechanism unsuitable for such scenarios.

Equally distributed feedback does not take the group size into account and is thus not affected by inaccurate group size estimations. While it provides the lowest feedback latency, it cannot prevent a feedback implosion at the sender.

For exponentially distributed feedback the number of feedback responses only varies within the limit of the statistical errors over several orders of magnitude. For groups with more than  $0.2 \cdot 10^6$  hosts, the values rise above the large plateau of the average fifteen response messages. This increase perfectly complies with the theoretical prediction. Below about 20-30 hosts, the large-group-limit does no longer apply and the observed number of responses drops below the theoretical value. The observed feedback latency drops exponentially with the number of group members. This behaviour was also expected from our analysis.

These advantageous properties hence recommend the exponential algorithm especially for scenarios with largely varying group sizes.

### 5.1 *Heterogeneous network latencies*

While it is safe to assume constant networks delays for satellite networks since the delay is almost solely determined by the propagation delay to and

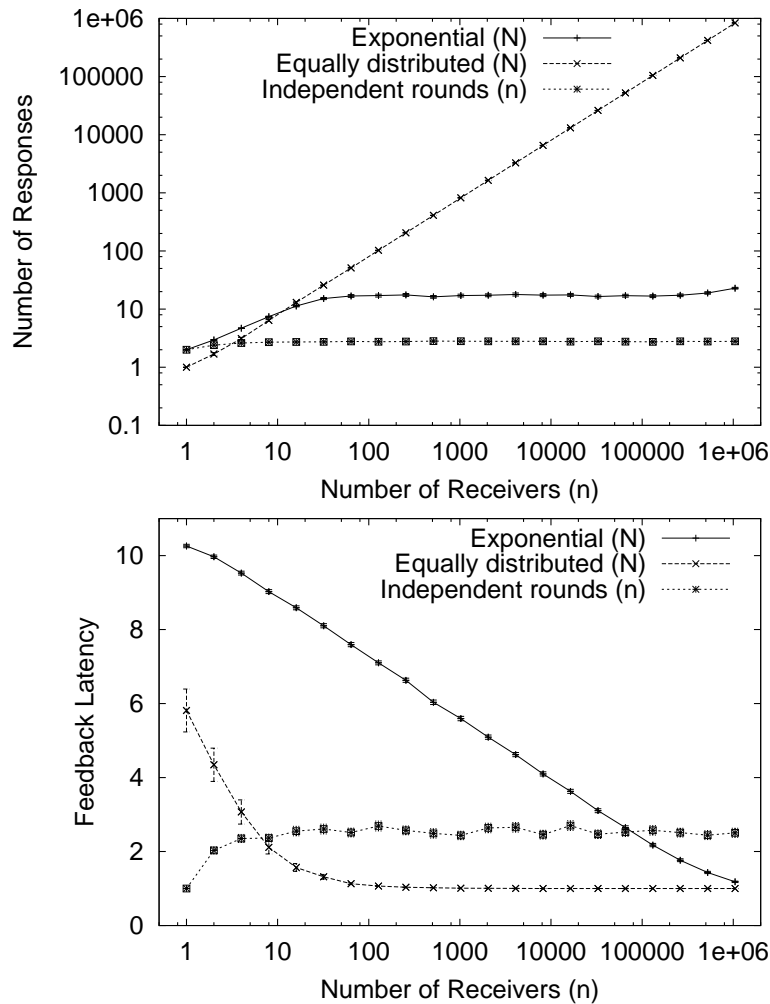


Fig. 5. Number of responses and feedback latency for the different feedback mechanisms

from the satellite, the assumption does not hold for terrestrial inter-networks. Simulation results to assess the impact of heterogeneous network delays are depicted in Figure 6. We assume a uniform distribution of the delays with delay variations ranging from 0% to 99%. The higher the variation in the network delays the lower the number of feedback messages, since more feedback can be suppressed by receivers with a low delay.

Heterogeneous network delays also significantly reduce the amount of feedback with the equally distributed feedback mechanism. However, heterogeneity has little impact when independent feedback rounds are used since the number of feedback messages is already very small.

For all mechanisms, the average feedback delay is slightly reduced by the delay variations. In our simulations, the feedback mechanisms thus profited from network heterogeneity.

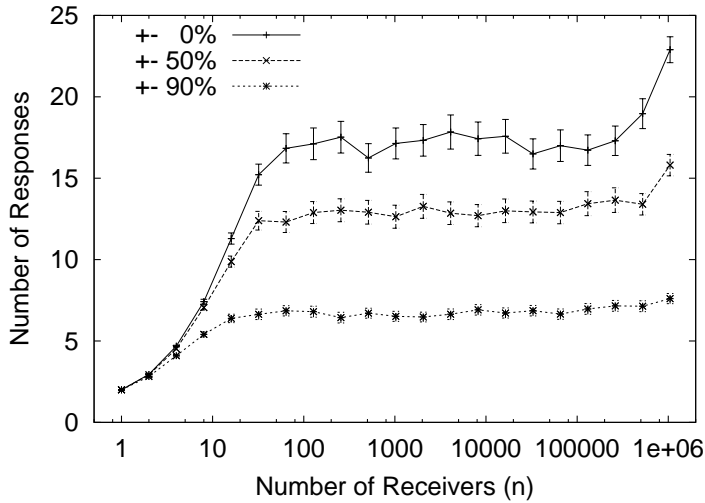


Fig. 6. Exponential feedback with heterogeneous network delay

## 5.2 Feedback rounds

If feedback is sent in rounds at discrete feedback times, the properties of the different mechanisms can be improved for environments with homogeneous network delays.

However, delay heterogeneity adversely affects the feedback properties. Comparing feedback responses of round-based and continuous feedback mechanisms for increasing delay variations reveals the deficiencies of round-based feedback. Figure 7 depicts the ratio of the number of responses with continuous exponential feedback and the number of responses with round-based feedback. For constant a network delay, the number of feedback messages can be reduced by up to a factor of 6 when using feedback rounds. Network delay variations of more than  $\pm 90\%$  reduce this ratio to 0.7, indicating that feedback rounds result in an increase of the number of responses at the sender. Thus, round-based feedback should only be used when the network latency is known and is fairly constant.

## 6 Related work

The necessity to employ scalable feedback algorithms in order to avoid feedback implosion has been obvious for a long time. Besides hierarchical [9,10,14] and token-based [3,4,18] approaches several random distributions have been studied: Floyd et al. [6,16] use equally distributed timers for their SRM (Scalable Reliable Multicast) protocol. The duration of the response interval is adopted according to the individual network latencies and the amount of response received. The latter method is inspired from various medium access



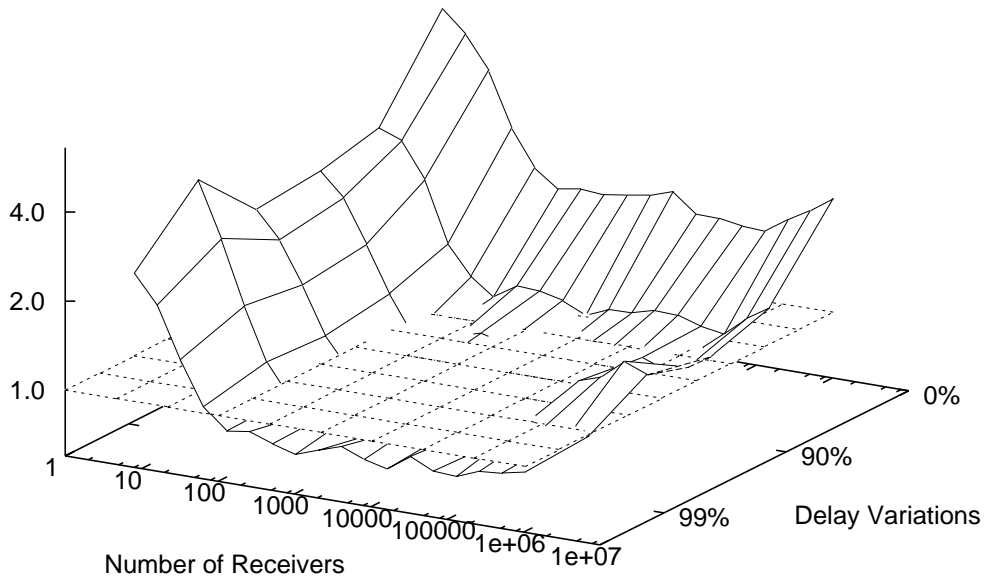


Fig. 7. Ratio of the number of responses with/without feedback rounds

protocols. Bolot, Turetli, and Wakeman [1] use an exponentially growing subspace of randomly assigned keys for their IVS video-conferencing system.

The recent advancements in the deployment of multicast in the Internet have further stimulated the interest in large multicast groups. Nonnenmacher and Biersack [13] study the statistical properties of three different timer distributions. Based on analytical and simulational results they derive optimised parameters for the algorithms and recommend the exponential feedback raise as most suited for large groups.

Lately some research has also been concerned with group size estimation based on feedback messages. Liu and Nonnenmacher [12] use the Poisson approximation for a maximum likelihood estimation of the group size. Friedman and Towsley [7] base their study on the binomial distribution.

Another direction of active research in the area of reliable multicast is the use of inner network nodes for feedback suppression [2,11]. Although this mechanism cannot be applied in the context of satellite networks on which we focused our study, feedback aggregation remains an important means to achieve scalability of multicast algorithms.

## 7 Conclusion

In this paper we have analysed three prototypical algorithms for feedback in very large multicast groups. We are especially interested in algorithms that are insensitive to a bad estimation of the actual group size. We have shown that only one of these algorithms, the exponential feedback raise, is able to provide sufficiently stable expected values across a large range of group sizes. Performing the feedback in rounds rather than continuously can further reduce the feedback latency slightly. On the other hand, feedback rounds require the adjustment of an additional parameter of the feedback algorithm to the network latency. Continuous feedback is thus superior if such additional knowledge is not available. Integrating our findings into the proposed algorithm we recommend to proceed as follows:

If the size of the multicast group is known and the network latency is known and rather constant, use the round-based feedback mechanism with constant feedback probability  $p$ .

Otherwise:

- (1) Estimate an upper limit  $N$  for the number of hosts that might provide feedback responses. This could be the number of hosts attached to the network.
- (2) Decide on the desired number  $R_0$  of feedback responses or the desired upper limit for the feedback latency  $T$ . Note that both values *cannot* be chosen independently since  $T = \tau \log_{R_0} N$  where  $\tau$  denotes the network latency (round-trip time for responses within the network).
- (3) Run the feedback-algorithm as follows:  
Upon reception of a feedback request each host that needs to send a response chooses a number  $x_i \in [0, 1)$  by conducting a Bernoulli experiment. If  $x_i < 1/N$  the respective host immediately sends a feedback-response. Otherwise it sends its response at time  $t_i = T(1 + \log_N x_i)$  unless it received a response confirmation before that time.
- (4) If a sufficiently good estimation for the network latency  $\tau$  can be given the following modification can be applied:  
The response interval  $[0, T]$  is divided into sub-intervals of duration  $\tau$ . Hosts that would respond within a given sub-interval send their response already at the beginning of the respective interval.

## Acknowledgements

This work has partly been funded by VIROR, the Virtual University Oberrhein, COST action 264 of the European Commission, and the RODEO group at INRIA, Sophia-Antipolis. Thanks to Emmanuel Duros, Rafael Rizzo, Thierry Turletti, and Walid Dabbous for valuable discussions. We would like to specially thank Martin Mauve, Jürgen Vogel, and Wolfgang Effelsberg who provided us not only with especially fruitful discussions but also convinced us to further study this interesting field.

## References

- [1] Jean-Chrystome Bolot, Thierry Turletti, and Ian Wakeman. Scalable feedback control for multicast video distribution in the Internet. In *Proceedings of the ACM SIGCOMM*, pages 58–67, 1994.
- [2] Brad Cain and Don Towsley. Generic multicast transport services: Router support for multicast applications. Technical report, University of Massachusetts, TR 99-74.
- [3] Jo-Mei Chang and Nick Maxemchuk. A broadcast protocol for broadcast networks. In *Proceedings of GLOBECOM*, page 19.2, December 1983.
- [4] Jo-Mei Chang and Nick Maxemchuk. Reliable broadcast protocols. In *ACM Transactions on Computer Systems*, volume 2(3), pages 251–273, August 1984.
- [5] Emmanuel Duros, Walid Dabbous, Hidetaka Izumiyama, Noboru Fujii, and Yongguang Zhang. *A Link Layer Tunneling Mechanism for Unidirectional Links*. Internet Engineering Task Force, June 1999. draft-ietf-udlr-lltunnel-02.txt.
- [6] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. A reliable multicast framework for light-weight sessions and application level framing. In *IEEE/ACM Transactions on Networking*, volume 5(6), pages 784–803, December 1997.
- [7] Timur Friedman and Don Towsley. Multicast session membership size estimation. In *Proceedings of IEEE INFOCOM*, pages 965–972, March 1999.
- [8] Thomas T. Fuhrmann. Protocol independent multicast and asymmetric routing. Technical Report 1/2000, Praktische Informatik IV, University of Mannheim, February 2000.
- [9] Matthias Grossglauser. Optimal deterministic timeouts for reliable scalable multicast. *IEEE Journal on Selected Areas in Communications*, 15(3):422–433, April 1997.

- [10] Markus Hofmann. A generic concept for large-scale multicast. In *Proceedings of the International Zurich Seminar on Digital Communication, LNCS*, volume 1044, pages 95–106, February 1996.
- [11] Sneha K. Kasera, Supratik Bhattacharyya, Mark Keaton, Diane Kiwior, Jim Kurose, Don Towsley, and Steve Zabele. Scalable fair reliable multicast using active services. *IEEE Networks Magazine*, January 2000.
- [12] Chuanhai Liu and Jörg Nonnenmacher. Broadcast audience estimation. In *IEEE INFOCOM 2000*. (to be published), March 2000. Tel Aviv, Israel.
- [13] Jörg Nonnenmacher and Ernst W. Biersack. Scalable feedback for large groups. In *IEEE/ACM Transactions on Networking*, volume 7(3), pages 375–386, June 1999.
- [14] Sanjoy Paul, Krishan K. Sabnani, John C. Lin, and Supratik Bhattacharyya. Reliable multicast transport protocol (rmtpt). In *IEEE Journal on Selected Areas in Communications*, volume 15(3), pages 407–421, April 1997.
- [15] Jonathan Rosenberg and Henning Schulzrinne. Timer reconsideration for enhanced RTP scalability. In *Proceedings of IEEE INFOCOM*, 1998.
- [16] Puneet Sharma, Deborah Estrin, Sally Floyd, and Van Jacobson. Scalable timers for soft state protocols. In *Proceedings of IEEE INFOCOM*, April 1997.
- [17] Jürgen Vogel. Entwurf und Implementierung eines generischen Late Join Mechanismus für interaktive Medien. Master’s thesis, Praktische Informatik IV, University of Mannheim, November 1999.
- [18] Brian Whetten, Todd Montgomery, and Simon Kaplan. A high performance totally ordered multicast protocol. In *Theory and Practice in Distributed Systems, Lecture Notes in Computer Science 938*. Springer-Verlag, 1994.