REIHE INFORMATIK

2/2000

On the Scaling of Feedback Algorithms for Very Large Multicast Groups

Thomas T. Fuhrmann
Praktische Informatik IV
University of Mannheim
68131 Mannheim, Germany

February 1, 2000

On the Scaling of Feedback Algorithms for Very Large Multicast Groups

Thomas T. Fuhrmann

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 give recommendations for the choice of well behaved feedback algorithms that are suitable for very large groups.

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.

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. With feedback latency we mean the expectation value for the time until the first response is sent. In section 2 we present a mathematical analysis of the feedback latencies of the three algorithms, and the number of response messages. Section 3 gives a comparison of these properties in the limit for large groups. In section 4 these analytical results are compared with results from simulations. Section 5 relates our work to other results obtained recently in this field of study. Section 6 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 that are currently being deployed or have been proposed recently. They all address the *at-least-one* scenario where a single response to a request suffices but multiple identical request from different group members will do no harm except for the superfluous network load. We assume that the request as well as the corresponding responses are multicast to the whole group. Hence, responses can be heard by other group members who then suppress own responses. For our first analysis we additionally assume a constant network latency between all group members. The question of packet-loss and heterogeneous network-latencies will be dealt with in the following sections.

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. Hence all of the three algorithms can be implemented as pure end-to-end protocols that do not interfere with inner network nodes. They are all based on random timers to pick early responders among the group members.

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[1] can be described as follows:

> *Let T be a constant upper time limit. Upon reception of a feedback request choose $x \in [0,1)$ at random and start a timer t. If a feedback response is heard 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 $i = 1 \ldots n$ are larger than x. Hence the probability that $x_{min} = \min\{x_1, \ldots, x_n\} \in [x, x+dx]$ is $n(1-x)^{n-1}dx$. The time corresponding to that choice of x is $t = xT$. Hence we obtain as expected feedback latency $L = < t_{min} >$

$$L = T \int_0^1 n(1-x)^{n-1} x \, dx = \frac{T}{n+1} \tag{1}$$

Since the feedback-responses are distributed equally over the interval T the expectation value for the number R of responses that are received is given by

$$R = n\frac{\tau}{T} \tag{2}$$

where τ is the network's latency.

[1] 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

Studying the algorithm just described 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 $t = 0$. An improved algorithm might thus perform feedback in rounds, i.e. it sends a certain number of responses only at the beginning of a feedback interval of length τ. No further responses are sent before the beginning of a eventual next round.

A straight-forward implementation of this idea can be phrased as follows:

> *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.

Since we discuss this method in more detail in the context of heterogeneous network latencies (Section 3.4) we will first analyse another possible realisation of the concept of feedback rounds [17]. It can be phrased as follows:

> *Let τ be the network's latency and $p \in (0,1]$ a constant. Upon reception of a feedback request randomly choose $x \in [0,1)$, 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 heard start a new round i.e. act as if another feedback request was received.*

Let n again be the number of hosts that can send a response. The number of feedback rounds can easily be calculated as $1 + (1-p)^n + (1-p)^{2n} + \cdots = \frac{1}{1-(1-p)^n}$. 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 expectation value for the feedback latency:

$$L = \tau \frac{(1-p)^n}{1-(1-p)^n} \tag{3}$$

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

$$R = \frac{np}{1-(1-p)^n} \tag{4}$$

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

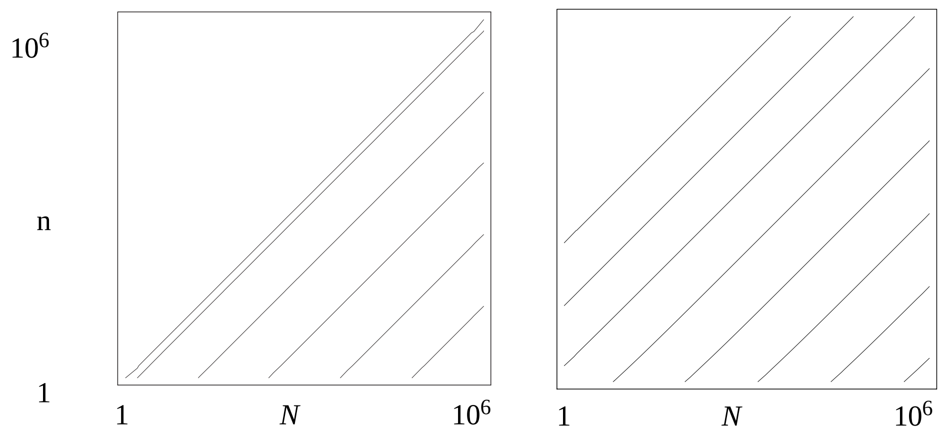


Figure 1: Feedback latency (left) and feedback responses (right) for independent feedback rounds (double-logarithmical plot). Contour lines *logarithmically* indicate values from $10^{-1}\tau$ to $10^{4}\tau$ and surplus responses from 10^2 to 10^{-5}.

2.3 Exponential feedback raise

A third alternative originally proposed by Bolot, Turletti, 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 the following modification of their mechanism:

> *Let N be the estimated number of group members and T be a constant upper time limit. Upon reception of a feedback request choose $x \in [0,1)$ and start a timer t. If a feedback response is heard 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 the first algorithm studied T is the guaranteed upper limit to the time when the first response is sent. To derive the expectation 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 $x_{min} = \min\{x_1, \ldots, x_n\} \in [x, x + dx]$ 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 expectation value for the earliest response:

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

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

Similarly, we can calculate the expectation 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 τ 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 expectation values for duplicate responses in these two cases

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

and

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

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

$$R = \int_0^{1/N}(n-1)\frac{N^{\tau/T-1}-x}{1-x}\cdot n(1-x)^{n-1}\,dx \tag{7}$$

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

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

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

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

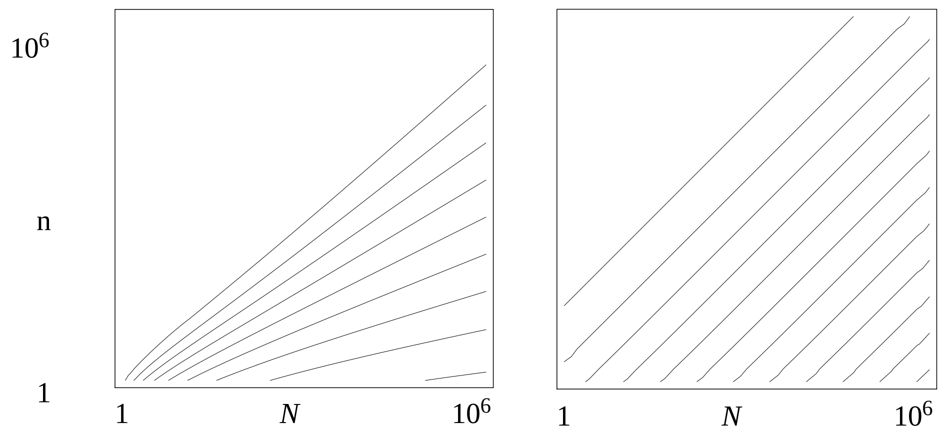


Figure 2: Feedback latency (left) and feedback responses (right) for Exponential Feedback Raise (double-logarithmic plot). Contour lines *linearly* indicate values from $0.1T$ to $0.9T$ and *logarithmically* indicate surplus responses from 10^2 to 10^{-9}.

3 Comparison of the algorithms' properties

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 a more thorough judgement 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 $R \geq nP(\tau)$. On the other hand $\forall \varepsilon > 0 : \lim_{n\to\infty} nP(\varepsilon) > 1$. Thus for $n \to \infty$ suppression immediately sets in at $t = \tau$ and we have $R = nP(\tau)$.

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 mostly be estimated rather easily. For example one can suppose 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 $\lim_{n\to\infty} R = \infty$ for fixed T we have to adapt T to the group size. Generally, by eliminating T we find a trade-off between feedback-latency L and response duplicates R:

$$\lim_{n\to\infty} LR = \tau \tag{11}$$

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 L and R. Thus this algorithm is not well suited for very large networks.

3.2 Independent feedback rounds

As already mentioned above, collecting responses for immediate sending at the beginning of each round can reduce the feedback latency by up to τ 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 $N \to \infty$

$$L_\infty = \tau \frac{e^{-\alpha}}{1 - e^{-\alpha}} = \frac{\tau}{e^{\alpha} - 1} \tag{12}$$

$$R_\infty = \frac{\alpha}{1 - e^{-\alpha}} \tag{13}$$

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 $L \simeq \frac{\tau}{\alpha+\cdots}$ while the number of feedback responses $R \to 1$.

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 $LR \simeq \frac{\tau}{\alpha}$ for $\alpha \ll 1$. Thus, this algorithm is as un-recommendable 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 $N \to \infty$ limit. Additionally, we choose T such that $N^{\tau/T} = e^{\beta} = \text{const}$, i.e. we set

$$\beta = \frac{\tau}{T} \ln N \tag{14}$$

With this adaption we now have

$$R_\infty = (\alpha + e^{-\alpha}) e^{\beta} \tag{15}$$

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

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

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 $\alpha \to 1$ case:

$$L_\infty \leq \frac{T}{\ln N} \int_{1/N}^{1} \frac{(1-x)^n}{1/N}\, dx \tag{17}$$

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

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

$$\frac{T}{10\log_{10}N} \leq L_\infty \leq T \tag{19}$$

Hence, unlike the two previous algorithms both expectation values remain rather insensitive to variations of α. Even in the limit $\alpha \to 0$ 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 expectation value of responses in the $\alpha \to 0$ limit by R_0 we have the following relation:

$$T = \tau \log_{R_0} N \tag{20}$$

3.4 Feedback rounds versus continuous feedback raise

As demonstrated the exponential feedback raise is superior to the other algorithms discussed here. For a thorough study it remains to be discussed whether the raise should be performed continuously or in rounds.

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 rounds mentioned in section 2.2 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

$$L = \int_0^T t \cdot n(1-P(t))^{n-1} P'(t)\, dt \tag{21}$$

$$= \int_0^T (1-P(t))^n\, dt \tag{22}$$

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 τ as compared to P_0 the feedback latency is generally reduced by ΔL 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 $\Delta L \simeq \frac{\tau}{2}$.

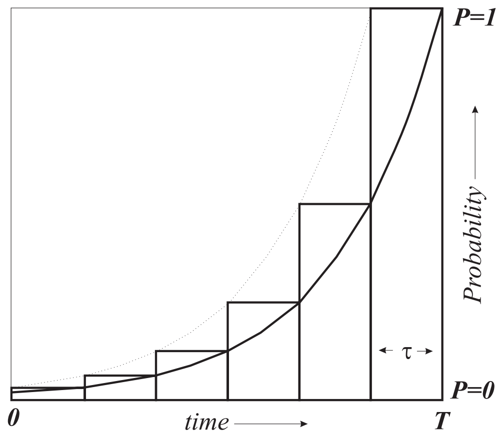


Figure 3: Exponential feedback with and without rounds.

If we however slightly underestimate the network latency τ 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 τ to be the maximal latency or a sub-group 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, τ 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.

Before concluding our analysis we shortly also mention 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 other participants received that feedback packet, a loss rate of p merely reduces the effective group size from n to $(1-p)n$. If additionally a fraction of q of the group received 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 does hence not principally affect the results given above.

4 Simulational results

In order to demonstrate the applicability of the exponential feedback raise algorithm proposed in section 2.3 a simulation model of this algorithm has been studied. For an upper limit of $N = 10^6$ hosts different groups with $n = 1$ to $n = 10^6$ actual hosts were studied. The results of these simulations are shown in Figure 4.

Both plots confirm the analytical findings discussed above:

1. Over several orders of magnitude the number of feedback responses only varies within the limit of the statistical errors. For groups with more than $0.2 \cdot 10^6$ hosts the values rise above the large plateau of the average four 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.

2. 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 this algorithm especially for scenarios with largely varying group size.

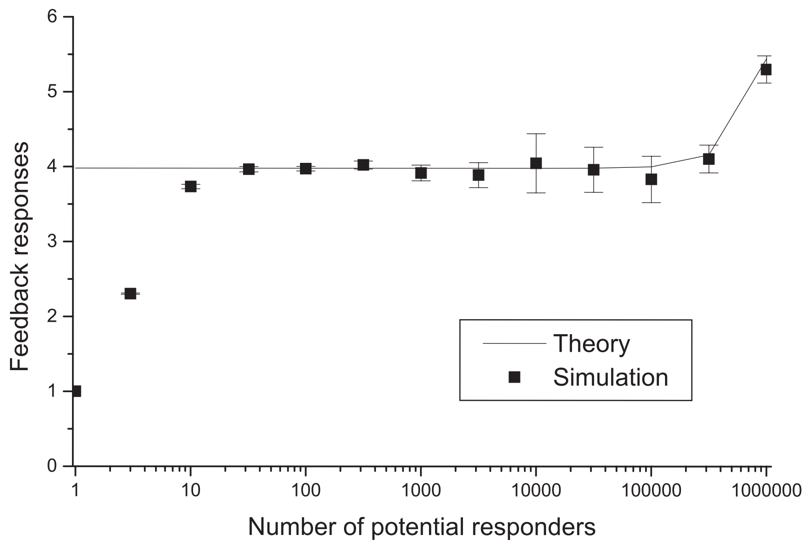
5 Related work

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 protocols. Bolot, Turletti, and Wakeman [1] use an exponentially growing sub-space 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



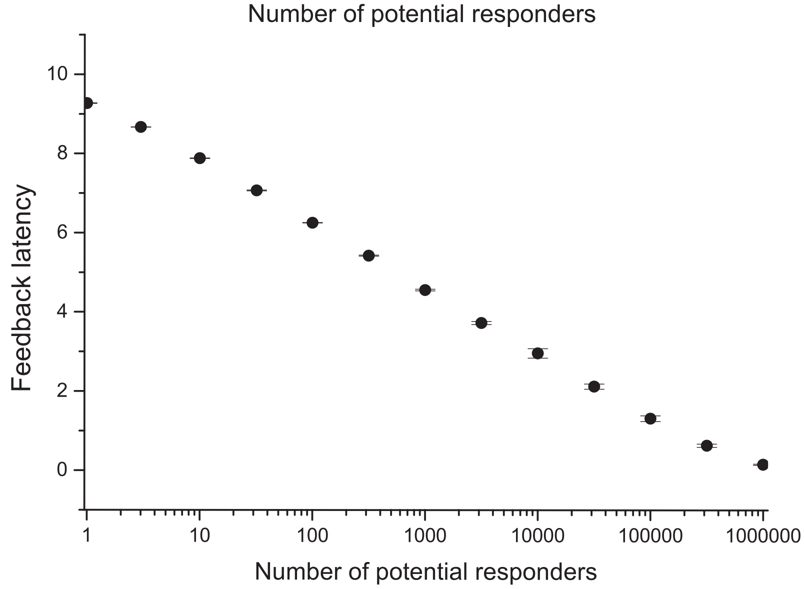


Figure 4: Comparison of analytic and simulational results of the number of feedback responses (upper plot) and of the feedback latency (lower plot) for $N = 10^6$

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.

6 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 expectation 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:

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 τ 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 $x_i \in [0,1)$. 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 from another host before that time.

4. If a sufficiently good estimation for the network latency τ can be given the following modification can be applied:
   The response interval $[0,T]$ is divided into sub-intervals of duration τ. 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. I would like to specially thank Martin Mauve, Jürgen Vogel, and Wolfgang Effelsberg who provided me not only with especially fruitful discussions but also convinced me 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 (rmtp). 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.