Improving Scalability on Reliable Multicast Communications

Daniel A. M. Villela [1]

Dept. of Electrical Engineering
Columbia University, USA

Otto Carlos M. B. Duarte [2]

Grupo de Teleinformática e Automação/COPPE/EE
Universidade Federal do Rio de Janeiro, Brazil

Abstract

This paper derives throughput and delay expressions for multicopy-based transmission on reliable multicast communications. The multicopy scheme consists of transmitting a fixed number of packet copies in order to increase the successful reception probability of packets and, consequently, decrease the required number of acknowledgements. For a large number of receivers, the results show a low mean delay and a high throughput. The multicopy strategy also presents a good performance under high error rates and heavy traffic load conditions.

Keywords: Reliable Multicast Protocols, Scalability, and Performance Evaluation.

1 Introduction

The performance bottleneck of communication systems has moved from bandwidth to packet processing time. In reliable multicast communication the processing overhead is even worse due to large number of acknowledgements to be processed. Scalability in multicast communications is essential, since many applications are being developed for a fast growing number of users. This is

[1] CNPq-Brazil Scholar
E-mail: dvillela@comet.columbia.edu
[2] E-mail: otto@gta.ufrj.br
This work was supported by UFRJ, FUJB, CNPq, and CAPES

the case of the MBone [1] that contains a number of such applications, e.g. whiteboard. High-speed networks and multimedia requirements increase the demand for distributed applications, which in turn require reliable, efficient, and scalable network multicasting service.

The reliability requirement implies the need for positive (ACK) or negative (NAK) acknowledgements to be returned to the sender. When the number of participants of a multicast session increases, the number of acknowledgements may become very high and generate an overload. This effect is called acknowledgement implosion and leads to an overhead in processing at the sender, in addition to a shortage of the various resources needed by the traffic. As a consequence, the quality of service (QoS) required by the applications cannot be achieved.

Solutions to reduce the number of acknowledgements and relieve acknowledgement implosion effects have been proposed. Several authors have proposed session hierarchies like tree and ring topologies [2–6] and local groups [7–9]. Nonnenmacher and Biersack [10,11], Rizzo and Vicisano [12,13] and Byers *et al.* [14] have proposed the use of Forward Error Correction (FEC) associated with the Automatic Repeat Request (ARQ) in an hybrid scheme. De Lima and Duarte [15] have analyzed an ARQ strategy with selective repeat for multicast communications. Floyd *et al.* [16] have introduced the Application Level Framing (ALF) concept in a Scalable Reliable Multicast (SRM) framework. Pingali *et al.* [17] have analyzed the throughput of reliable multicast protocols and have compared sender-initiated protocols and receiver-initiated protocols with and without a NAK suppression mechanism [18,16]. Yamamoto *et al.* [19] have used these models for delay analysis. Another delay analysis was derived by Villela and Duarte [20], with the introduction of a priority scheme that ensures a more stable performance of data transmissions and scalability.

This paper analyzes the use of multicopy transmission to reduce the impacts of acknowledgement implosion. Multicopy transmissions have been previously introduced by Sastry [21] for satellite protocols. The proposed scheme consisted of a Go-Back-N protocol that retransmits repeated copies of a packet when a NAK is received. Birrel [22] has analyzed a multicopy scheme for bursty transmission. De Lima and Duarte [23] have also analyzed a multicopy scheme for satellite protocols. These schemes, however, have been intended for different purposes and several questions have not been addressed for current scenarios, for instance the Internet.

This work, unlike the previous ones, analyzes multicopy transmissions concerning scalability. In [19], the delay of single-copy protocols is analyzed in relation to the load imposed to the network. Besides the traffic load, analysis on this paper is mainly based on number of receivers and it provides several comparisons between multicopy and single-copy cases. For throughput several

similar comparisons are taken and different number of copies are considered. In addition, results are also compared to FEC-based protocols, particularly using fast erasure codes as Tornado codes [14]. The obtained results show that the multicopy strategy presents high performance for high packet loss rate within the network or large multicast groups. Moreover, important contributions are analyzing suitable number of copies to be used and quantifying performance enhancements over single-copy protocols and over well-know mechanisms like NAK avoidance.

This paper is structured as follows. Section 2 describes the multicopy strategy and its motivation and defines the two protocols analyzed. In Section 3 the delay and the throughput equations are derived. Section 4 shows the obtained numerical results. Finally, conclusions are given in the Section 5.

2 Multiple Copies in Reliable Multicast Communications

This paper considers the scheme of multicopy transmission in the context of reliable multicast communications. The scheme is proposed for a high loss rate environment of multicast communications like the current Internet. An important result to be achieved is scalability, i.e, high throughput under a large number of multicast receivers.

2.1 The Transmission of Multiple Copies

Participants of reliable multicast communications send data packets and wait for acknowledgements or timeouts before declaring a packet lost or successfully transmitted. Positive acknowledgements are returned by receivers to indicate the successful delivery of a packet. Otherwise, after a predefined timeout without receiving acknowledgements the packet is assumed lost and the sender must retransmit it. A protocol may use negative acknowledgements to request the retransmission of specific lost packets. If packet losses occur at a low rate, one transmission is enough for correct reception of most of the data packets. In the ideal case, when there is no loss, a single transmission suffices for every data packet to be received by all receivers. However, the scenario is quite different if the packet loss rate is high, in which case the number of transmissions required for every receiver to correctly receive a packet is high as well and can grow with the increase of the number of receivers.

Figure 1 shows the number of necessary transmissions for the correct transfer of a single packet to all receivers. With large sets (approximately 10,000 receivers and loss probability 0.25), more than seven transmissions are required.

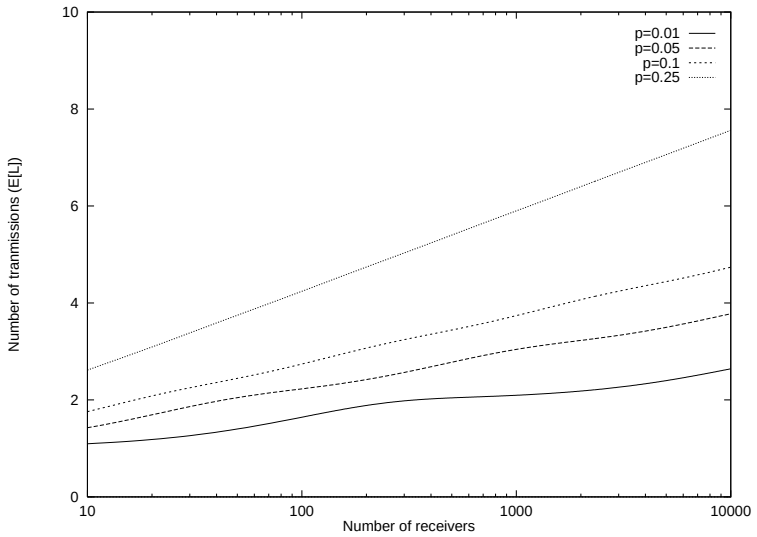


Figure 1. Required transmissions to correct transfer a packet to the whole set of receivers.

With the sending of multiple copies, the number of returned NAKs is expected to diminish. In the same way, because the sending of additional copies of a packet is no longer prompted by a returning NAK, the overall delay is also expected to diminish.

The key idea of the multicopy transmission scheme is to increase the probability of a successful packet transfer. At every transmission, the sender transmits m copies instead of a single one. This increases the probability of a successful packet transfer. Furthermore, the replication of packets is very simple to implement and requires little additional processing.

The sending of multiple packet copies consumes more bandwidth than the sending of single copy. Under low error rate conditions, this use of additional bandwidth can hardly be justified. On the other hand, if error rates are high, the transmission of negative acknowledgements is more likely to occur and may lead to acknowledgement implosion. We show in this paper that under these circumstances, the requirement of additional bandwidth to send multiple copies is by far compensated by the lower delays and higher throughput in the correct reception by all receivers.

FEC schemes were proposed to enhance reliable multicast protocols. Both FEC and multicopy make use of redundant data to enhance reliable multicast communications. After proposal of Reed-Solomon codes in reliable multicast protocols[10], Tornado codes were proposed [14] because they are faster, although Tornado codes are slightly more inefficient. This inefficiency exists because more than the packet group size is required to reconstruct an entire group size. Reed-Solomon codes, however, are optimal because they need the same group size to reconstruct the original packets. We also show in this paper

that even for fast codes as Tornado the encoding time is still large when compared to the overall mean packet-processing time and causes lower processing rates as well as more delay.

2.2 Multicopy Protocols

In this paper the multicopy scheme is applied to NAK-based protocols. In the first protocol to be defined, a receiver that detects a packet loss asks for its retransmission with a negative acknowledgement. Retransmission is achieved following selective repeat model. Furthermore, first packet retransmission occurs upon NAK receiving, but in case of subsequent retransmissions of the same packet, they are triggered by timers.

Another protocol is defined by adding to the first one a NAK-avoidance mechanism as proposed by Ramakrishnan and Jain [18]. Whenever the receiver detects a packet loss it waits for a random time interval before returning the NAK. In this case, the receiver multicasts its NAK to the other receivers as well. Whenever those receivers realize that a NAK related to that packet has already been requested, they refrain from sending their own NAKs.

Before moving into the next section, it is necessary to present some important remarks. Whenever a multicopy-based protocol is mentioned, the first protocol is being referred, unless noticed explicitly the use of NAK avoidance. The first protocol can be also referred by **RM** (**R**eceiver-initiated with **M**ultiple Copies) and the second one, **RMS** (**R**eceiver-initiated with **M**ultiple Copies and NAK **S**upression) for simplicity (specially in equations and figures). Single-copy protocols are a special case in which the number of transmitted copies is equal to one.

3 Performance Evaluation

This section analyzes the performance of multicopy-based multicast protocols. For performance evaluation, a model is created based on a set of assumptions. The loss probability of a data packet, p_r, is the same for all receivers. Further, losses are independent in time and within a group of copies and acknowledgements are never lost. A single data source is considered and packets are sent to receivers following a Poisson distribution with rate λ. The propagation delay, τ, between the sender and each of the receivers is the same for all participants in the multicast system. Without loss of generality, the model assumes that the multicast group is composed of R receivers. Similar assumptions have also been used in [17,19].

Random variables X and Y denote the processing time of data packets and acknowledgements, respectively. These random variables have means $E[X]$ and $E[Y]$ and second moments $E[X^2]$ and $E[Y^2]$. The processing of replicas is very simple and usually incurs little additional cost, because several functions, like header processing, are only executed once. Thus, its implementation becomes much easier and the processing is faster than those of more complex algorithms, e.g., FEC coding. That is an advantage of this strategy and, for this reason, the processing time of the m copies can be taken to be a random variable $X_g = X + f\sum_{i=1}^{m-1} X$, where f $(0 \leq f \leq 1)$ is a coefficient that represents the reduction of the processing time for one replica. Therefore,

$$E[X_g] = E[X] + f(m-1)E[X]. \tag{1}$$

Usually, f is close to zero. By the independence of p_r for all packets, the loss probability of all m data-packet copies is $p = p_r^m$. The number of transmissions required for the correct reception of a data packet by receiver i is a random variable $N_i^{(r)}$. Its probability density function can be written, for $n \geq 1$, as

$$P(N_i^{(r)} = n) = (1-p)p^{n-1}, \tag{2}$$

and its mean, $E[N_i^{(r)}] = 1/(1-p)$.

Applying the same reasoning to all receivers, the number, N, of required transmissions for the correct reception by all receivers is related to $N_i^{(r)}$ by the expression

$$N = \max_{1 \leq i \leq R} N_i^{(r)}. \tag{3}$$

Its probability density function and its average are

$$P(N \leq n) = P(N^{(r)} \leq n)^R = (1-p^n)^R \tag{4}$$

and

$$E[N] = \sum_{n=1}^{\infty}(1-(1-p^{n-1})^R). \tag{5}$$

3.1 Delay Analysis

These quantities are useful in the following delay analysis of the multicopy-based protocols. Delay is the time elapsed from packet arrival at sender's buffer to the complete and correct reception by all receivers [19].

In order to obtain the delay, two phases can be distinguished. First, there is a time interval, D_{RM}^{P1}, necessary to detect the loss and, later, another time interval, D_{RM}^{P2}, needed to request the retransmission and to wait for the complete reception [20].

For a receiver randomly chosen, the probability that at least one packet in an m-copy group is not lost is $1-p$. With the reception of at least one packet, the delay is the sum of propagation delay, processing times and waiting times at both receiver and sender. On the other hand, the probability that all packets in an m-copy group are lost is p and the delay is the sum of the elapsed periods of time on the two described phases. Thus, the total expected delay for this system is obtained:

$$\begin{aligned} E[D_{RM}] = &(1-p)(E[W_{RM}^{S}] + E[W_{RM}^{R}] + E[X] + E[X_g] + \tau) \\ &+p(E[D_{RM}^{P1}] + E[D_{RM}^{P2}]). \end{aligned} \tag{6}$$

If NAK suppression is introduced, analysis becomes more complex. However, the operation remains almost the same as before (two main phases) and the protocol delay is found with similar analysis. The main difference comes from the random time interval introduced by the NAK avoidance, denoted by a random variable V. This interval occurs between the two phases. The expression for average value of V was derived in [19].

Then, the average delay is derived in this case:

$$\begin{aligned} E[D_{RMS}] = &(1-p)(E[W_{RMS}^{S}] + E[W_{RMS}^{R}] + E[X] + E[X_g] + \tau) \\ &+pE[V] + p(E[D_{RMS}^{P1}] + E[D_{RMS}^{P2}]). \end{aligned} \tag{7}$$

3.2 Processing Rate Analysis

The NAK implosion problem creates a limitation for processing rate of the multicast systems. As the number of receivers grows the number of NAKs to be processed gets even higher and further increases the processing delay. Whenever delay gets higher, the achieved throughput gets lower. This section outlines a maximum throughput analysis of multicopy protocols.

The protocols' throughput is given by the minimum of processing rates at sender and receivers. The total processing times are random variables, called T_s and T_r, and the processing rates, Λ_s and Λ_r, for sender and receiver, respectively.

Basically, for computing mean processing times, it is necessary to consider

the time to process new transmissions and their group of copies, possible retransmissions, NAKs and timers. The expressions for a multicopy protocol are derived (more details in the appendix) as follows:

$$1/\Lambda_s^{RM} = E[T_s^{RM}] = E[N]E[X_g] + E[L^{RM}]E[Y] \tag{8}$$

and

$$\begin{aligned} 1/\Lambda_r^{RM} = E[T_r^{RM}] = & mE[N](1-p_r)E[X] + E[(N_r-1)^+]E[Y] \\ & + E[(N_r-2)^+]E[X_t] \ , \end{aligned} \tag{9}$$

where L^{RM} is a random variable denoting the number of packet losses until its successful reception by all receivers and X_t is another random variable corresponding to timer processing time.

When considering NAK suppression, the expressions for processing rates are obtained in a similar fashion. The NAK suppression mechanism will reduce the number of NAKs generated from the previous case. By increasing a random delay before NAK returning, the expected number of NAKs is often one. If there is a suitable choice of parameters such as the random delay necessary for the mechanism operation, the situations in which more than one NAK is returned will be unusual [17]. Without taking into account this situation, the expressions for processing rate at sender and receivers are, respectively,

$$1/\Lambda_s^{RMS} = E[T^{RMS}] = E[N]E[X_g] + (E[N]-1)E[Y] \text{ and} \tag{10}$$

$$\begin{aligned} 1/\Lambda_r^{RMS} = E[T^{RMS}] = & mE[N](1-p_r)E[X] + (E[N]-1)E[Y] \\ & + \frac{p^2}{1-p}E[X_t]. \end{aligned} \tag{11}$$

Finally, the throughput is equivalent to minimum of both processing rates (for both cases):

$$\Lambda^P = min\{\Lambda_s^P, \Lambda_r^P\}, \text{ (P=RM or RMS).} \tag{12}$$

4 Numerical Results

In [9], realistic values for packet processing times were found for a networking environment consisting of UDP/IP protocols and Linux operating system. Their normalized values are adopted in order to compute quantitative delay and processing rate results for multicopy protocols. That is, $E[X]$ assumes the

value 1.0, $E[Y]$, 0.2 and the propagation delay, $\tau = 20.0$. The considered loss probability ranges from 0.01 to 0.25 as observed in Mbone traffic [24].

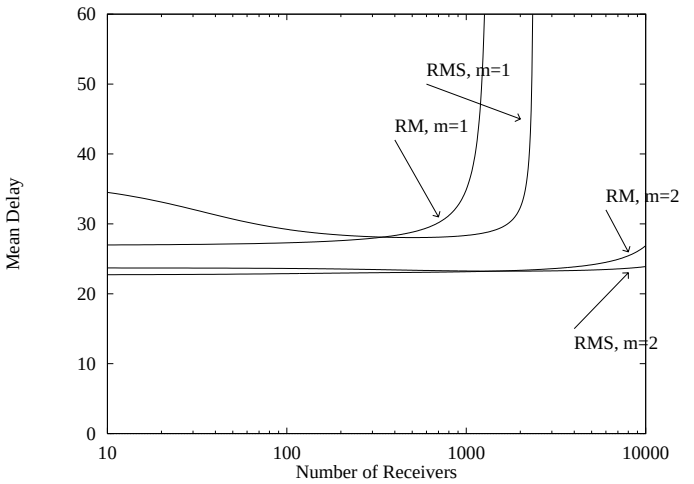
Figure 2(a) depicts expected delay for multicopy-based protocols, with packet loss probability $p_r = 0.1$ and data generation rate $\lambda = 0.1$. While the single-copy can only support up to 1,000 receivers, before delay increases sharply, when copies are used (for $m = 2$) the delay changes slightly within the observed range (from 10 to 10,000 receivers). The point in which a sharp increase of delay occurs is hereby called a performance breakup point. Only near 10,000 receivers the mean delay starts to increase. When there is NAK suppression in the system, even near 10,000 receivers the delay does not significantly increase.

It can be observed from Figure 2(a) that the mean delay decreases under the NAK avoidance mechanism over a range of few receivers. This is explained since the random time interval imposed by the mechanism is smaller when the number of receivers gets higher. Only after certain number of receivers, the influence of the random time will get smaller, the processing will affect the performance and NAK implosion occurs.

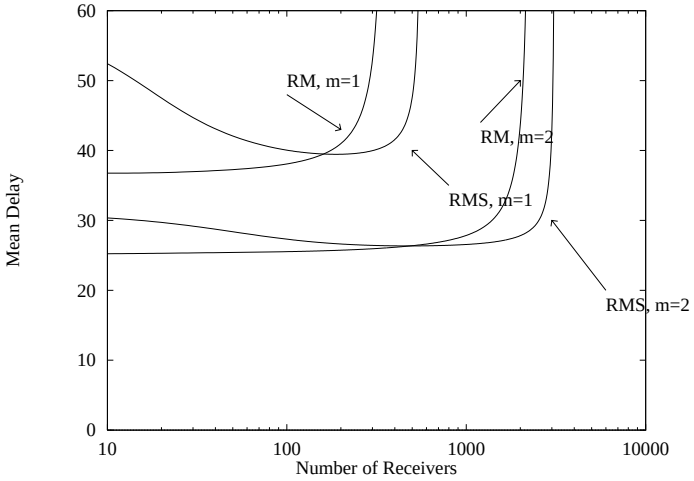
Figure 2(b) illustrates the case for a loss probability $p_r = 0.25$. The same comparisons can be made but on a different degree, since the loss rate is higher. For instance, the number of receivers to be supported is smaller. The system presents smaller stability range than the preceding case with $p_r = 0.1$. For multicopy-based protocols performance breakup point occurs near 2,000 and 3,000 receivers, respectively. The effect of increasing the number of receivers supported due to NAK avoidance is again noted clearly. As the loss rate is higher in this case, the random time interval will have a stronger influence. Once again we observe that multicopy reduces the delay. The ratio between delay experimented by the simple multicopy protocol under $m = 1$ and $m = 2$ (around 1.48) for few receivers shows the delay reduction due to duplicated transmissions.

Performance enhancements by multicopy are more effective than by NAK avoidance. The boundaries for stability in Figure 2(b) quantify the enhancements resulted from multicopy and NAK avoidance. A comparison between single-copy protocols without and with NAK suppression leads to a two times better performance for the latter. For multicopy, comparing the simplest multicopy protocol with the simplest single-copy protocol, the range of receivers is enhanced by five times (from 200, for $m = 1$, to 1,000, for $m = 2$).

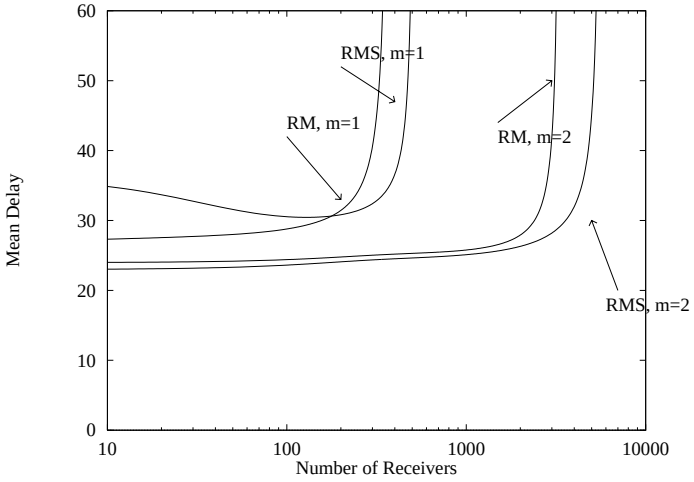
Another interesting fact is the reduction of number of acknowledgements by the NAK suppression mechanism and its implication on scalability. This was already known, but it can be analyzed quantitatively. The NAK suppression mechanism supports larger number of receivers with stability. The reduction of NAKs returned to the sender causes less processing for the participants



(a)



(b)



(c)

Figure 2. Delay of multicopy and single-copy protocols for (a) $p_r = 0.1$ and $\lambda = 0.1$, (b) $p_r = 0.25$ and $\lambda = 0.1$ and (c) $p_r = 0.1$ and $\lambda = 0.2$.

of the multicast communications and, consequently, for a higher number of receivers, NAK implosion will not be experienced. The results from Figure 2(b) supports this observation.

Figure 2(c) depicts average delay under loss probability $p_r = 0.1$, but with packet generation rate $\lambda = 0.2$. The purpose is to investigate the impact of increasing traffic load on system performance. It is expected that the increase in number of packets is associated with increase of NAKs, and as a result a possible NAK implosion. The ratio previously discussed between delays observed of the simple multicopy-based protocol under $m = 1$ and $m = 2$ for low number of receivers remains with the same values. This is a consequence of copies transmitted and loss rate, as the number of receivers has no critical influence under this situation. What is most important to note is that the packet-generation rate increase can affect scalability and multicopy protocols are able to perform better under such conditions.

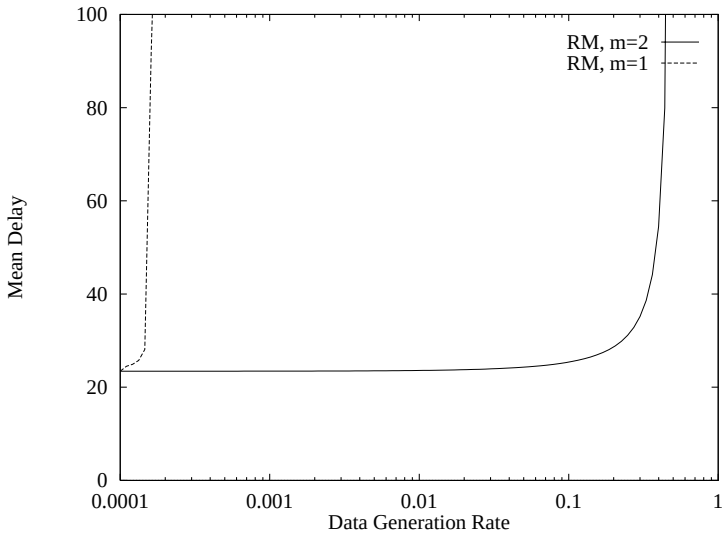
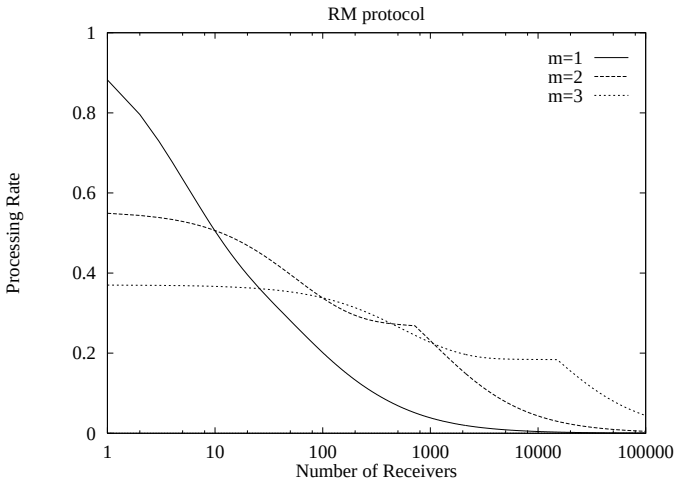


Figure 3. Observed delay for 10,000 receivers.

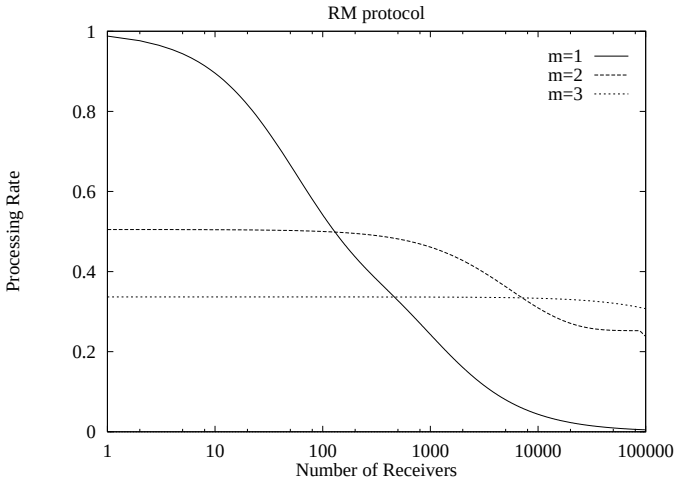
As data generation rate increases, the delay also increases, but making use of two copies enhances significantly the performance in a scenario with a large number of receivers. As seen on Figure 3, for 10,000 receivers the sustainable data generation rates are quite different. By simply using two copies, the data generation rate is bounded by 0.2, while for a single copy, the maximum value for stability is between 0.0001 and 0.0002. Thus, the multicopy protocol has a performance boundary three orders greater than the single-copy one.

Following the analysis of delay, quantitative results are drawn for processing rates according to the performance evaluation and using the same average processing times of packets and NAKs.

Figure 4(a) shows the throughput of NAK-based protocol for integer values of



(a)



(b)

Figure 4. Protocol processing rate for $m = 1, 2, 3$ and: (a)$p_r = 0.1$, (b)$p_r = 0.01$.

m from 1 to 3, loss probability of 0.1 and coefficient $f = 0.1$. The throughput achieved for a single-copy case ($m = 1$) under a low number of receivers (approximately one receiver) is 0.9. However, this throughput decreases quickly, as far as the number of receivers increases. In the range comprising 1 to 10 receivers, the processing rate remains approximately constant around 0.5. On the other hand, using multicopy with $m = 3$, from 1 receiver to 100 receivers, the processing rate also remains unchanged and its value is approximately 0.333. These observations were expected, since the protocol transmits a group of m copies by each packet, the expected throughput is a fraction of $1/m$ of the throughput with a single packet. Figure 4(a) shows another important effect related to scalability. As for $m = 2$ and $m = 3$ the throughput decreases more slowly than for $m = 1$, the throughput under a large number of receivers with multiple copies is larger than with single packets. Figure 4(b) depicts these effects for packet loss probability $p_r = 0.01$. For $m = 2$ an almost constant

throughput in the range $1 - 1,000$ and a slow decrease for higher number of receivers is noticed. For $m = 3$ and on the whole range $1 - 100,000$ the rate remains in a tight range and the observed decrease of throughput is very small.

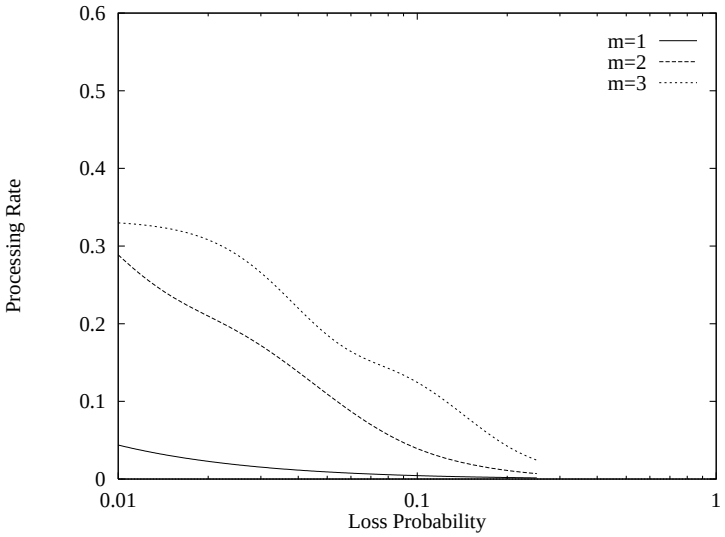
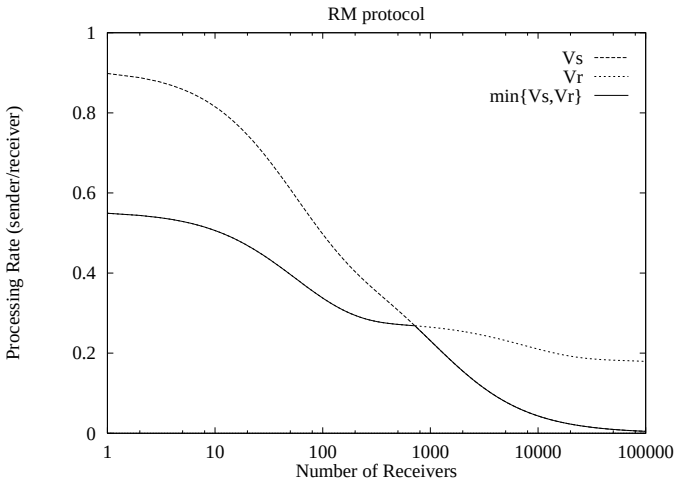


Figure 5. Protocol processing rate for 10,000 receivers.

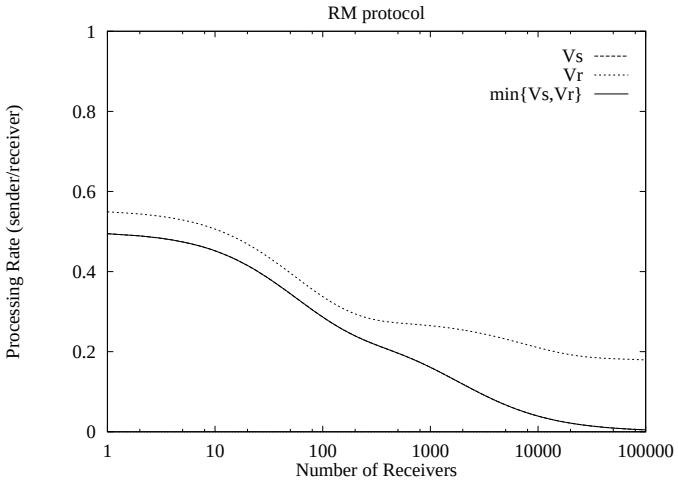
Figure 5 depicts protocol processing rate under 10,000 receivers. As observed, for a large set of receivers the multicopy brings performance enhancements. The processing rates achieved with $m = 2$ and $m = 3$ far exceeds the processing rate achieved for $m = 1$. This strengths the conclusion that multicopy results in better scalability than single-copy protocols. The combined use of single-copy and multicopy depending on number of receivers, i.e., using single-copy simplification for a small-number of receivers is recommendable.

Furthermore, curves in Figure 4(a) are not totally well behaved because they are sharp-pointed for certain number of receivers. This effect happens since the processing rate is the minimum of rates achieved in both sender and receivers. Figure 6(a) shows these rates for loss probability 0.1 and $m = 2$. Close to 1,000 receivers there is a point where the minimum of the rates changes from receiver rate to sender rate. The parameter $f = 0.1$ will cause a sender processing rate larger than receiver processing rate from one receiver to this point. Moreover, the sender processing rate for a low number of receivers is approximately 0.9. As copies of packets take only a fraction of mean packet processing time, the sender rate is not decreased by a factor $1/m$. Nevertheless, if a value one is assigned to coefficient f, this rate will certainly diminish by this factor. This is shown in Figure 6(b), constructed with the same parameters but changing coefficient f to value one.

Figure 7 presents protocol processing rate, when using NAK suppression, for $m = 1, 2$, and 3, loss probability 0.1 and coefficient $f = 0.1$. Once more for



(a)



(b)

Figure 6. Protocol processing rates at sender and receivers for: (a) $p_r = 0.1$ and $f = 0.1$, (b) $p_r = 0.1$ and $f = 1$.

a low number of receivers, the throughput achieved by the m-copy protocol scheme is a fraction $1/m$ of the processing rate when single packets are used. Nevertheless, as the number of receivers increases, the three curves get much closer.

An interesting option is done considering the coefficient $f = 1$. This is equivalent to consider that every copy will require the same mean time to process the packet at senders. Figure 8 depicts the three curves with m =1, 2,3 respectively, for $p_r = 0.1$ as the case of Figure 4(a), except using coefficient $f = 1$. The throughput for $m = 2$ decreases much slower than for single copy packet. In the case of 3 replicas the processing rate presents the slightest decrease.

Multicopy enhances the performance and causes less necessity of NAK avoid-

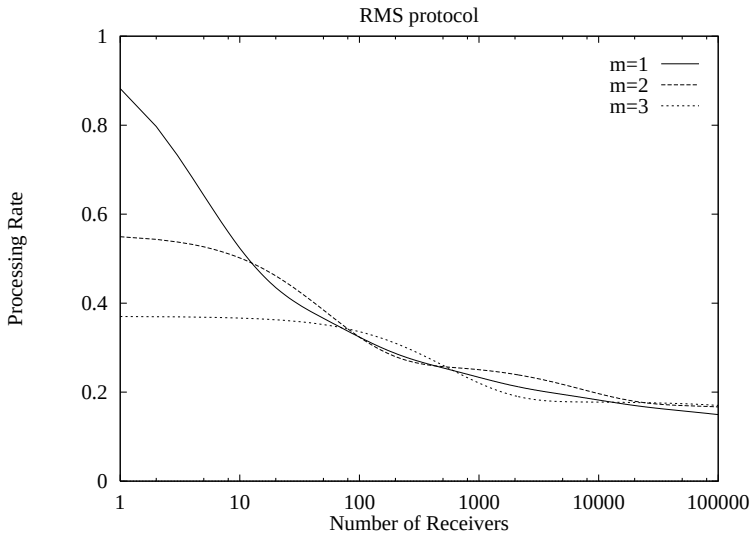


Figure 7. Protocol processing rate for $p_r = 0.1$, $f = 0.1$, $m = 1, 2, 3$ and using NAK avoidance.

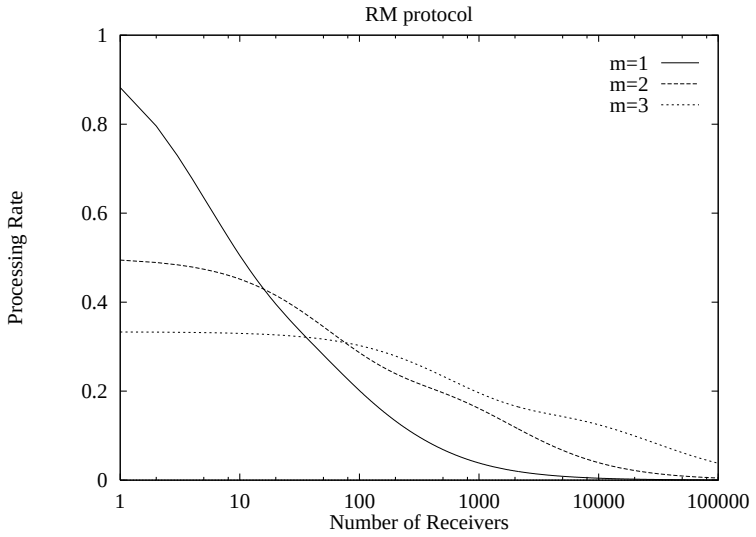
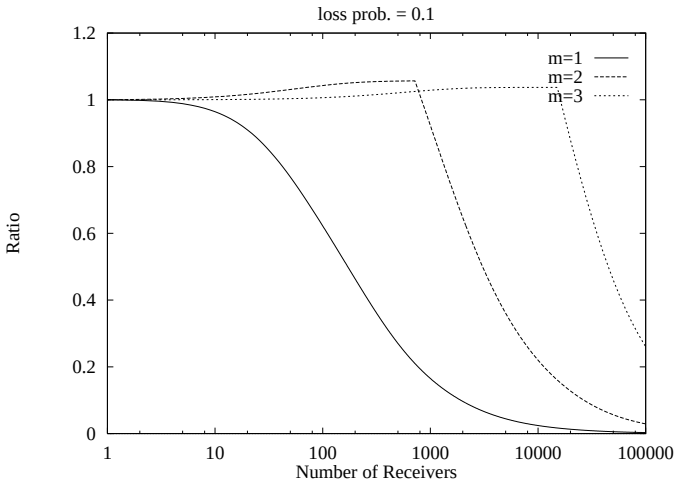


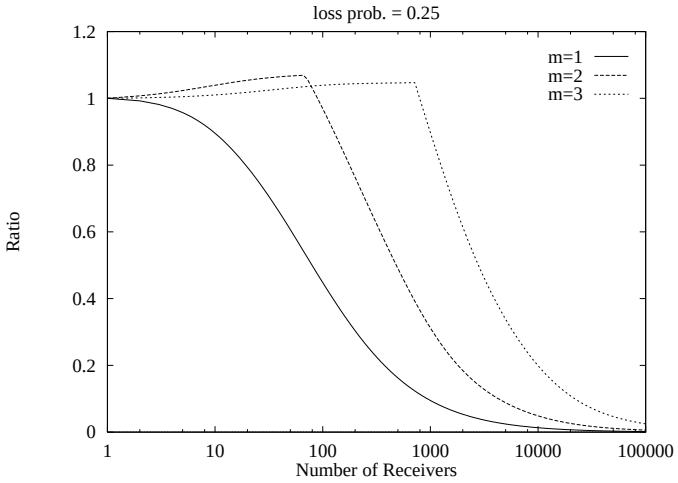
Figure 8. Protocol processing rate for $f = 1$, $p_r = 0.1$ and $m = 1, 2, 3$.

ance. Since NAK suppression will cause higher throughput of systems with large number of receivers, in comparison to the case without suppression, it is also interesting to study the ratio between the throughput of both protocols. The Figures 9(a), 9(b), 9(c) show the ratio $\Lambda_{RM}/\Lambda_{RMS}$ for loss probabilities 0.1, 0.25 and 0.01, respectively. It is shown that for a wide range of receivers the ratio maintains one. As m increases the range is wider as well. On the other hand, as loss probability increases, the range becomes smaller. For $m = 3$ and $p_r = 0.01$, the ratio remains constant in the whole observed range. Thus, multicopy reduces the performance difference between the cases with and without NAK suppression.

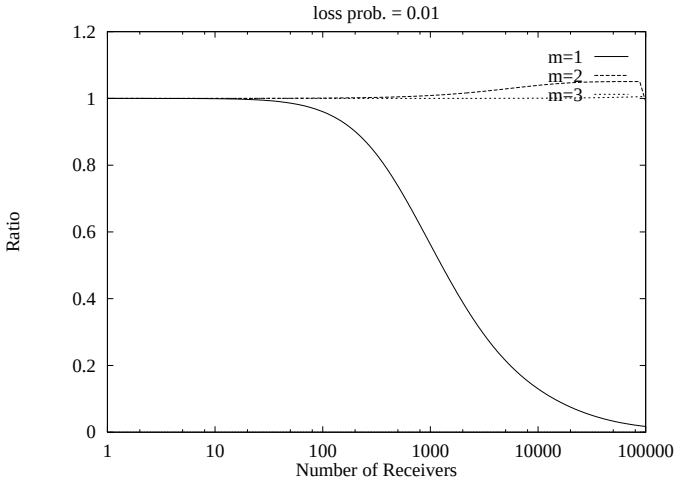
Figure 10 depicts a comparison of processing rates for the use of Tornado Z



(a)



(b)



(c)

Figure 9. Ratio $\Lambda_{RM}/\Lambda_{RMS}$ with same values of m and $f = 0.1$: (a)$p_r = 0.1$, (b)$p_r = 0.25$, (c)$p_r = 0.01$.

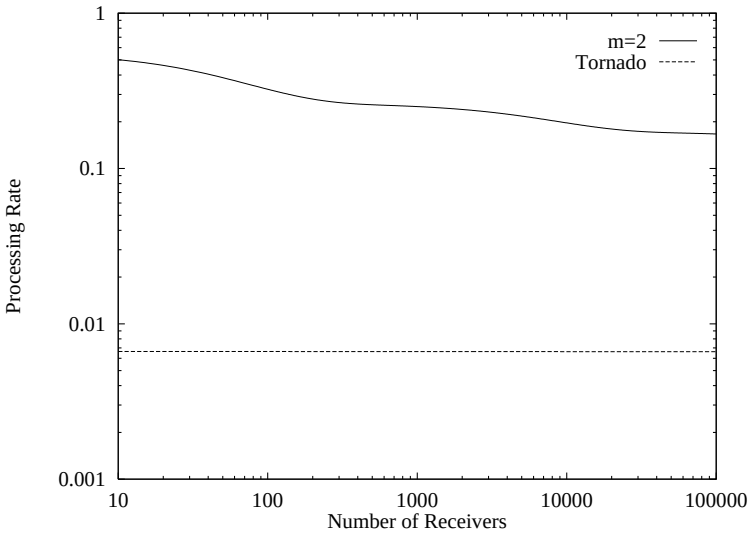


Figure 10. Processing rates for protocols using Tornado Z codes and multicopy ($m = 2$).

codes and multicopy. The FEC-based protocol considered here is very similar to the one shown in [10], excepting that a Tornado code is used instead of Reed-Solomon codes. To this protocol, a group of packets with size k is transmitted. NAKs are returned to the sender when losses are noticed (using NAK avoidance) and redundant packets are transmitted to reconstruct original packets on the receiver side. In Figure 10, the size of a packet group is 500 packets, value of f is 0.1, loss probability is 0.1 and $m = 2$ for the multicopy protocol. In order to compare the two of them, the NAK avoidance scheme is applied to both. The analysis done by Nonnenmacher and Biersack [10] was adapted to reflect the use of Tornado codes, since that analysis was considering Reed-Solomon codes. In order to find the encoding and decoding constants of the Tornado Z, the benchmark results shown in [14] were used. The average of the encoding and decoding times was considered after taking into account the encoding and decoding proportional to $(k + l)log(1/\epsilon)P$, where k is the number of packets, l is the number of redundant packets, ϵ is the measurement of inefficiency of the erasure codes and P is the packet size. In Figure 10 it was considered an optimal solution using the Tornado Z, i.e., by knowing the average number of rounds necessary to reconstruct the group and taking this number and the Tornado inefficiency ϵ into account to limit the number of packets to encode. In [14] it was used a number of redundant packets equal to the actual number of packets, which is very costly. If not optimized, the processing rate can be smaller by 3 orders of magnitude.

Even though multicopy protocols require more bandwidth, we observe that the multicopy scheme presents processing rate 75 times greater than the Tornado-based protocol for 10 receivers and 25 times greater for 100,000 receivers. In fact the encoding time is the bottleneck for the FEC protocol. It was found 75ms the time for encoding 500 packets, while the mean processing time for

a packet and copies ranges from 613 μs (for 10 receivers) to 1.9 ms (100,000 receivers).

Intuitively, the encoding time will always impose an initial delay in order to encode packets. Also, the inefficiency of Tornado codes increases the number of rounds necessary to receive the transmission group since $(1+\epsilon)k$ packets are necessary to decode and reconstruct the original packets. This can also lead to higher delays, considering the propagation delay for each retransmission round.

Therefore, encoding times are dominant and reduce processing rates. In fact this effect was identified in [10], and the given solutions included to preencode the packets off-line which will demand larger storage for data or to enhance the hardware at the sender. Figure 10 shows that multicopy is an alternative for scalable environments and does not require such changes.

Finally, figure 11 shows the processing rate for several values of m for $p_r = 0.1$ and $f = 1$. It can be noted that the increase of the value of m leads to a higher throughput when a large number of receivers is considered. Nevertheless, the maximum throughput decreases with m and therefore, high values of m are not recommended. For the current MBone traffic conditions, the m-copy strategy fits well for $m = 2$ and $m = 3$.

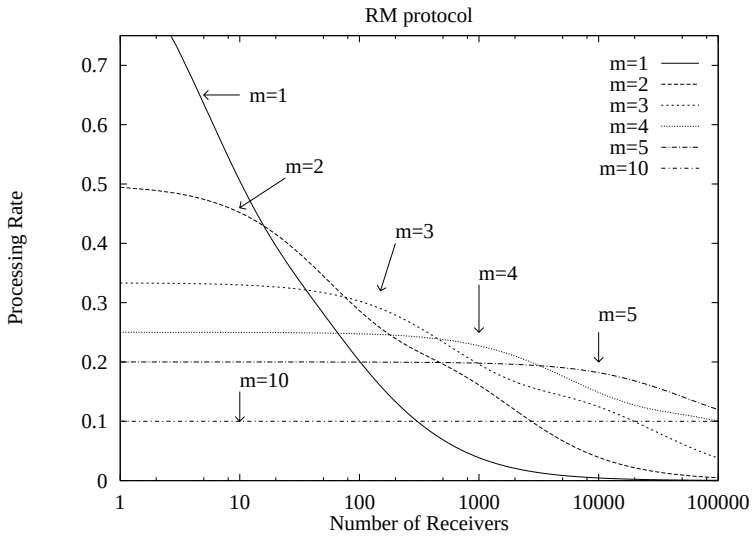


Figure 11. Processing rate for $p_r = 0.1$, $f = 1$ and $m = 1, 2, 3, 4, 5$ and 10.

5 Conclusions

This paper has analyzed the delay and throughput of a multicopy scheme applied to reliable multicast communications. The multicopy transmission

scheme increases the successful transfer probability of a packet and, consequently, decreases the required number of retransmission requests. Thus, this strategy reduces the effects of acknowledgement implosion for a large number of receivers. Moreover, it reduces also the packet transmission delay, since copies of every packet are sent in advance.

This work has analyzed multicopy-based protocols concerning scalability. Based on a scalability purpose the multicopy strategy has been compared to the single-copy scheme. Also, this work has considered delay of multicopy-based protocols and quantified it by increasing the number of receivers besides the traffic load.

The throughput of multicopy-based protocols has also been derived and several comparisons have been made based on packet loss probability, number of copies and number of receivers. Under some studied cases, the results for multicopy with large number of receivers far exceeds the performance of a single-copy scheme. Also, the results have pointed out suitable values of the number of copies to be used.

Multicopy is simpler than FEC schemes which still have costly encoding and decoding times if compared to protocol processing times.

Finally, in some depicted situations the performance enhancement using multicopy is greater than adopting a NAK-suppression mechanism. In some circumstances (loss rate 0.25 and data generation rate 0.2) on analyzing the delay, the boundary of number of receivers using multicopy is enhanced by five times, whereas adopting NAK avoidance doubles the number. The throughput difference between cases with and without NAK suppression gets smaller when working under the multicopy strategy.

The performance is enhanced using multicopy under high packet loss rates and for large sets of receivers. Therefore, multicopy adoption is recommended for scalability purposes.

Acknowledgements

The authors would like to thank Valmir Barbosa, Raymond Liao and Artur Ziviani for their helpful comments on the text. Daniel Villela is thankful to Conselho Nacional de Desenvolvimento Tecnológico/Brazil for supporting his PhD study at Columbia University (ref. 200168/98-3).

References

[1] H. Eriksson, "MBONE: The multicast backbone," *Communications of the ACM*, vol. 37, pp. 54–60, Aug. 1994.

[2] J. Nonnenmacher and E. W. Biersack, "Performance modelling of reliable multicast transmission," in *Proc IEEE INFOCOM'97*, (Kobe), Apr. 1997.

[3] B. N. Levine and J. J. Garcia-Luna-Aceves, "A comparison of known classes of reliable multicast protocols," in *Proceedings of International Conference on Network Protocols*, (Columbus), Oct. 1996.

[4] J. C. Lin and S. Paul, "RMTP: A reliable multicast transport protocol," in *Proc. IEEE INFOCOM'96*, pp. 1414–1424, Mar. 1996.

[5] J. Liebeherr and B. S. Sethi, "A scalable control topology for multicast communications," in *IEEE INFOCOM'98*, Apr. 1998.

[6] Baghwat, Mishra, and Tripathi, "Effects of topology on performance of reliable multicast communication," in *Proc. IEEE INFOCOM'94*, pp. 602–609, 1994.

[7] M. Hofmann, "Adding scalability to transport level multicast," in *3rd COST 237 Workshop - Multimedia Telecommunications and Applications*, (Barcelona), Nov. 1996.

[8] H. W. Holbrook, S. K. Singhal, and R. D. Cheriton, "Log-based receiver-reliable multicast for distributed interactive simulation," in *Proceedings of SIGCOMM '95*, (Cambridge MA), Aug. 1995.

[9] S. Kasera, J. Kurose, and D. Towsley, "Scalable reliable multicast using multiple multicast groups," in *Proceedings of 1997 ACM Sigmetrics Conference*, 1997.

[10] J. Nonnenmacher and E. W. Biersack, "Parity-based loss recovery for reliable multicast transmission," in *Proc. SIGCOMM'97*, (Cannes), Sept. 1997.

[11] J. Nonnenmacher, E. W. Biersack, and D. Towsley, "Reliable multicast: Where to use forward error correction," in *Proc. 5th Workshop on Protocols for High Speed Networks*, (Sophia Antipolis, France), Oct. 1996.

[12] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Computer Communication Review*, vol. 27, pp. 24–36, Apr. 1997.

[13] L. Rizzo and L. Vicisano, "RMDP: a FEC-based reliable multicast protocol for wireless environments," *ACM Mobile Computer and Communication Review*, vol. 2, Apr. 1998.

[14] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. of ACM Sigcomm'98*, Sept. 1998.

[15] H. M. de Lima and O. C. M. B. Duarte, "An effective selective repeat ARQ strategy for high speed point-to-multipoint communications," in *IEEE Global Telecommunications Conference GLOBECOM'96*, pp. 1059–1063, Nov. 1996.

[16] S. Floyd, V. Jacobson, S. McCanne, C. G. Liu, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *ACM Computer Communication Review*, pp. 342–356, Aug. 1995.

[17] S. Pingali, D. Towsley, and J. Kurose, "A comparison of sender-initiated and receiver initiated reliable multicast protocols," *IEEE Journal on Selected Areas in Communications*, vol. 15, Apr. 1997.

[18] S. Ramakrishnan and B. N. Jain, "A negative acknowledgement with periodic polling protocol for multicast over LAN," in *Proc. IEEE INFOCOM '87*, pp. 502–511, Mar. 1987.

[19] M. Yamamoto, J. F. Kurose, D. F. Towsley, and H. Ikeda, "A delay analysis of sender-initiated and receiver-initiated reliable multicast protocols," in *Proc. of IEEE INFOCOM'97*, 1997.

[20] D. A. M. Villela and O. C. M. B. Duarte, "A priority scheme applied to scalable reliable multicast communications," in *Proc. of IEEE GLOBECOM'98*, (Sydney), pp. 3368–3373, Nov. 1998.

[21] A. R. K. Sastry, "Improving repeat-request (ARQ) performance on satellite channels under high error rate conditions," *IEEE Transactions on Communications*, vol. COM-23, pp. 436–439, Apr. 1975.

[22] N. D. Birrel, "Preemptive retransmission for communication over noisy channels," *IEEE Proceedings*, vol. 128, pp. 393–400, Nov. 1981.

[23] H. M. de Lima and O. C. M. B. Duarte, "A go-back-n protocol with multicopy retransmission for high speed satellite communications," in *Proc. of the IEEE International Conference on Communications SUPERCOMM/ICC'94,*, (New Orleans), pp. 859–863, May 1994.

[24] M. Yajnik, J. Kurose, and D. Towsley, "Packet loss correlation in the MBone multicast network," in *IEEE Global Internet Conference*, (London), Nov. 1996.

[25] L. Kleinrock, *Queueing Systems Vol. 1: Theory.* Wiley Interscience, 1974.

Appendices

A Delay Analysis

The delay analysis of multicopy protocols follows the same reasoning considered in [19] for multicast protocols.

The means for $E[D_P^{P1}]$ and $E[D_P^{P2}]$ are:

$$E[D_P^{P1}] = \sum_{k=0}^{R-1} \binom{R-1}{k} p^k (1-p)^{R-k-1} \frac{p^{k+1}/\lambda}{1-p^{k+1}} + E[W_P^S] + E[W_P^R] + \tau + E[X] + E[X_g] + E[Y]$$

$$\text{and } E[D_P^{P2}] = \frac{p(T_R + E[W_P^R] + E[Y])}{(1-p)} + E[W_P^S] + E[W_P^R] + E[X] + E[X_g] + 2\tau, \tag{A.1}$$

where W_P^R, W_P^S are random variables for waiting time on buffers of receiver and sender, respectively, and $P = RM$ or RMS.

In order to obtain the waiting times on buffers it is important to identify flows at both sender and receivers. Focusing first on the simple multicopy protocol, the sender processes data and acknowledgments. Data arrives at rate λ_t^S and acknowledgments comprise not only NAKs that cause retransmissions but also NAKs that are processed but do not cause retransmissions. The arrival rates for these NAKs are, respectively,

$$\lambda_r^S = \lambda(E[N] - 1) \tag{A.2}$$

and

$$\lambda_n^S = \lambda(RE[N^{(r)}] - E[N] - R + 1). \tag{A.3}$$

The m data packets have processing time according to the random variable X_g. The NAKs that cause retransmission have processing time $X_g + Y$ and the other NAKs simply Y. Thus, the total load at senders is

$$\rho^S = \lambda E[N]E[X_g] + \lambda R(E[N^{(r)}] - 1)E[Y]. \tag{A.4}$$

As the data generation rate is assumed to be a Poisson process, the queuing system at the buffer can be seen as an M/G/1 queue. Therefore, the mean waiting time at the sender is given by the Pollaczec-Kinchin formula [25],

$$E[W_{RM}^S] = \frac{\lambda_t^S E[X_g^2] + \lambda_r^S (E[X_g^2] + E[Y^2] + 2E[X_g]E[Y]) + \lambda_n^S E[Y^2]}{2(1-\rho^S)} \tag{A.5}$$

Under this protocol, two flows are present at the receivers. A first flow consists of received data and the other one is composed by NAKs generated by the receivers themselves. The service times are, respectively, X and Y. As the number of transmitted replicas is m and the distribution for packets received

within a group is binomial, the mean number of packets received in a group is $m(1-p_r)$. Therefore, the arrival rates for data and NAKs are, respectively,

$$\lambda_d^R = m\lambda E[N](1-p_r) \tag{A.6}$$

and

$$\lambda_n^R = \lambda(E[N^{(r)}]-1) \tag{A.7}$$

and the total load observed at receivers is

$$\rho^R = m\lambda E[N](1-p_r)E[X] + \lambda(E[N^{(r)}]-1)E[Y]. \tag{A.8}$$

Thus, the mean waiting time at the buffers of receivers is

$$E[W_{RM}^R] = \frac{\lambda_d^R E[X^2] + \lambda_n^R E[Y^2]}{2(1-\rho^R)}. \tag{A.9}$$

A.1 NAK avoidance

Considering NAK avoidance, at sender's buffer there are again data and acknowledgments. The acknowledgments are also composed by two kinds of NAKs: the ones that cause retransmission and the other ones that do not. However, the number of received NAKs decreases due to NAK avoidance mechanism. A random variable Q denotes the number of received NAKs. The total NAK arrival rate occurs not only at sender but also at other receivers is $\lambda E[Q]$, since NAKs are multicast. The arrival rates of data and of NAKs (two different flows) are

$$\lambda_t^S = \lambda,\ \lambda_r^S = \lambda(E[N]-1) \text{ and } \lambda_n^S = \lambda(E[Q]-E[N]+1), \tag{A.10}$$

with service times X_g, $X_g + Y$ and Y, respectively.

Load and waiting time are found by the relationships described in Appendix A substituting the new arrival rates. The expression for $E[Q]$ has been derived in [19].

At receivers, data arrival rate is $\lambda_t^R = mE[N](1-p_r)$. Thereby, the load is $\rho^R = m\lambda E[N](1-p_r)E[X] + \lambda E[Q]E[Y]$.

Consequently, the mean waiting time of packets in buffers, W_{RMS}^R, is easily found:

$$E[W_{RMS}^R] = \frac{\lambda_t^R E[X^2] + \lambda E[Q]E[Y^2]}{2(1-\rho^R)}\ . \tag{A.11}$$

B Processing Rate Analysis

The processing rate analysis is similar to the analysis derived for multicast protocols analyzed in [17]. The total processing time can be expressed as a sum of two processing times. A first one is composed by necessary time to process transmissions of a new packet and its group of copies, and the possible necessary retransmissions. Another processing time will be generated by the received NAKs. As processing time for one packet is being called X_g and for a NAK, Y, the total processing time for this packet is

$$T_s^{RM} = \sum_{n=1}^{N} X_g + \sum_{i=1}^{L^{RM}} Y, \tag{B.1}$$

where L^{RM} is a random variable that corresponds to number of times that the packet is lost until its correct reception by all receivers, which is also the number of NAKs returned to the sender.

Thereby, the average processing time is

$$E[T_s^{RM}] = E[N]E[X_g] + E[L^{RM}]E[Y]. \tag{B.2}$$

To obtain the expression for $E[L^{RM}]$, the number of NAKs returned by one randomly chosen receiver should be obtained first. This value corresponds to $(N^r - 1)^+$, where the function $f(x) = (x)^+$ is defined as $(x)^+ = max(0, x)$. Its mean is simply $E[(N^r - 1)^+] = p/(1-p)$. For all R receivers, as receivers are statistically equal, $E[L^{RM}]$ is simply $Rp/(1-p)$.

Therefore, the total processing time is expressed by

$$E[T_s^{RM}] = E[N]E[X_g] + E[L^{RM}]E[Y], \tag{B.3}$$

and the sender throughput is $\Lambda_s^{RM} = 1/E[T_s^{RM}]$.

The processing time at receivers must be also evaluated. By the same reasoning, a random variable T_r^{RM} is assigned to the total processing time. For a successful reception of one packet there are a processing time to receive correctly the entire group of m copies that corresponds to $mE[N](1-p_r)E[X]$. The required processing for creating the negative acknowledgments and returning them to the sender is $E[(N_r - 1)^+]E[Y]$. There is another processing component for receivers related to expiration of timers, whose mean time is $E[(N_r - 2)^+]E[X_t]$, where X_t is the random variable corresponding to processing time of a timer.

The expression for $E[(N_r-1)^+]$ has been already evaluated and $E[(N_r-2)^+]$ is easily obtained: $E[(N_r-2)^+]=p^2/(1-p)$. Thus, the total processing time and, consequently, the receiver processing rate is reached:

$$1/\Lambda_r^{RM} = E[T_r^{RM}] = mE[N](1-p_r)E[X] + E[(N_r-1)^+]E[Y] + E[(N_r-2)^+]E[X_t]. \tag{B.4}$$

Finally, the protocol throughput will be the minimum of the processing rate at sender and receivers:

$$\Lambda^{RM} = min\{\Lambda_s^{RM}, \Lambda_r^{RM}\}. \tag{B.5}$$

Using NAK avoidance processing rates are obtained in a similar fashion.