Scalability Enhancement of Reliable Multicast Communications by Using Multicopy

Daniel Antunes Maciel Villela
daniel@gta.ufrj.br

Otto Carlos Muniz B. Duarte*
otto@gta.ufrj.br

Grupo de Teleinformática e Automação – GTA
COPPE/EE - Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
P. O. Box 68504 - 21945-970 - Rio de Janeiro - RJ - Brasil
http://www.gta.ufrj.br

Abstract

This paper analyzes a strategy of multicopy transmissions to be applied to reliable multicast communications. This strategy consists on the transmission of fixed number of copies of a packet in order to guarantee its successful reception and, consequently, decrease the number of acknowledgments. Throughput and delay analyses for two multicast protocol schemes are derived. The results show a stable behavior of the mean delay and a high throughput for a large number of receivers. The multicopy strategy presents a good performance even at high error rate and heavy traffic generation conditions.

Keywords: Reliable Multicast Protocols, Scalability and Performance Evaluation.

1 Introduction

Scalability in multicast communications is essential, since many applications are being developed, where the number of participants is growing in a high rate. The MBone [1] is at such a platform and contains a number of these applications like whiteboard, for instance. High speed networks and multimedia needs increase the demand for distributed applications. Thus, there is a set of

*This work was supported by UFRJ, FUJB, PROTEM-CC, CNPq, CAPES and COFECUB.

requirements for a class of transport protocols that will provide this service. They must be reliable, efficient, and scalable.

The requirement of reliability implies positive (ACK) or negative (NAK) acknowledgments that are returned to the sender. When the number of participants of a multicast communication increases, the number of acknowledgments may become very high and generate an overload. This effect is called acknowledgment implosion and leads to overhead in sender processing and lack of resources needed by the traffic. Therefore, the quality of service (QoS) required by applications can not be achieved.

Solutions to diminish the number of acknowledgments and reduce the effects related to implosion problem have been proposed. Some authors have proposed hierarchies as tree and ring topologies [2, 3, 4, 5, 6] and local groups [7, 8, 9]. Nonnenmacher *et al.* have proposed the utilization of Forward Error Correction (FEC) associated with the Automatic Repeat Request (ARQ) in an hybrid scheme [10, 11, 12, 13]. De Lima and Duarte [14] have analyzed an ARQ strategy with selective repeat for multicast communications. Floyd *et al.* [15] have introduced the Application Level Framing (ALF) concept. Pingali *et al.* [16] 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 [17, 15]. Yamamoto *et al.* [18] have used these models for delay analysis. Another delay analysis was derived by Villela and Duarte [19] with the introduction of a priority scheme that ensures more stability for performance of data transmissions and their scalability.

Multicopy transmissions have been previously introduced by Sastry [20] for satellite protocols. The proposed scheme was a Go-Back-N protocol that retransmits repeated copies of a packet when a NAK is received. Birrel [21] has analyzed a multicopy scheme for saturated transmission. De Lima and Duarte [22] have also analyzed a multicopy scheme for satellite protocols.

This paper analyzes the multicopy transmission to reduce effects caused by acknowledgment implosion. The multicopy scheme is applied to the generic protocols presented by Pingali *et al.* in [16]. Both delay and throughput are analyzed considering variation of number of receivers, loss rate, data generation rate and number of generated copies. The analytical results show that the multicopy strategy is robust and suitable for systems which have scalability as important goal.

This paper is structured as follows: the section 2 describes the multicopy strategy and its motivation and defines the two analyzed protocols. In section 3.1 the delay performance evaluation is derived and numerical results are obtained. Section 3.2 shows throughput analysis and quantitative results are presented. Finally, the section 5 has the final conclusions.

2 Multiple Copy in Reliable Multicast Communications

This paper considers the scheme of multicopy transmission to be used in reliable multicast communications. This scheme is applied to generic protocols that operate with well-known acknowledgement mechanisms presented on previous works [17, 15, 16]. Their operations are briefly discussed in this section.

2.1 The Transmission of Multiple Copy

Participants of reliable multicast communications send data packets and wait for acknowledgements or timeouts to consider a packet lost or successfully transmitted. Positive acknowledgements are returned by the receivers to indicate the successful transmission of a packet. Otherwise, after a predefined timeout without receiving acknowledgments the packet is assumed lost and the sender must retransmit it. A protocol may use of negative acknowledgments to ask the retransmission of lost packets. If the packet losses occur in a low rate, one transmission will be enough for correct reception of most of the data packets. On the ideal case, when there is no loss, every data packet is received with only one transmission.

Nevertheless, the scenario is quite different if the packet loss rate is high. The number of transmissions required to every receiver correctly receive a packet is high and becomes higher with the increase of the number of receivers.

Figure 1 shows the number of necessary transmissions for a correct transfer of a packet to all receivers. With large sets (approximately 10,000 receivers), more than seven transmissions are required. Thus, whenever m copies are sent, the packets will be sent in advance and the number of returned NAKs will diminish. Further, the forward transmission also causes a reduction of the experimented average delay, since the time necessary for NAKs returning will not happen and the NAK processing will be much smaller.

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

Although the multiplicity of packets consumes bandwidth, if it is required more than one transmission for complete reception by all receivers, the packets that would be sent in retransmissions, are sent in advance. An incorrect reception of a packet means the transmission of negative acknowledgment and, for a great number of receivers, this means acknowledgment implosion. The tradeoff is extra bandwidth demand (number of copies) against correct reception by all receivers without retransmission.

2.2 Multicopy Protocols

The multicopy scheme is applied to the receiver-initiated protocols defined in [16, 18]. The protocols with single packet transmissions are simply produced by assigning m to value one. Thus, a receiver

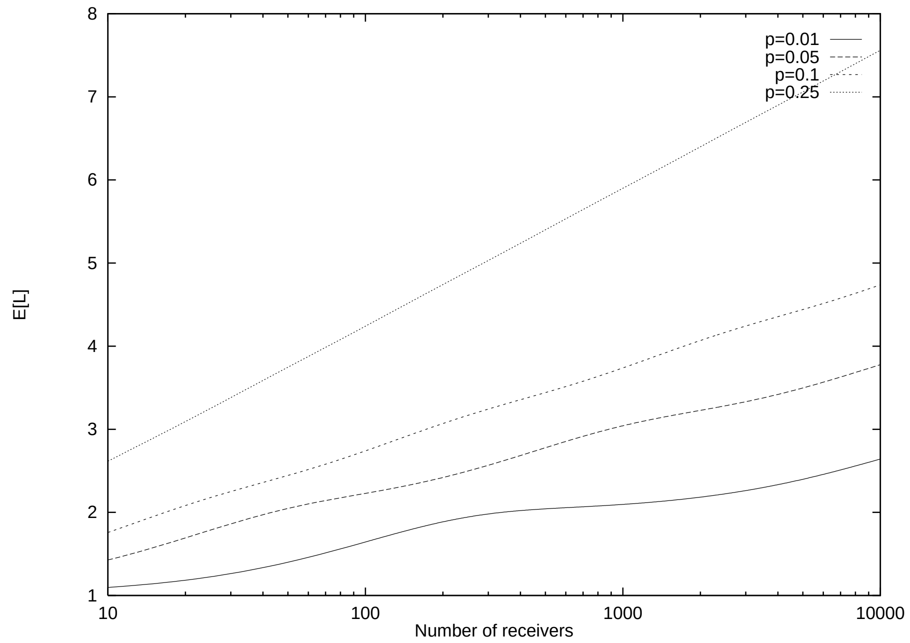


Figure 1: Required transmissions to correct transfer a packet to the whole set of receivers.

that detects a packet loss will ask for its retransmission with a negative acknowledgement. The adopted retransmission fashion in the defined protocol is selective repeat. This protocol will be named **RM** (**R**eceiver-initiated with **M**ultiple Copy). Its procedures can be summarized:

- The sender multicasts m data-packet replicas to the receivers.
- Whenever a receiver detects a packet loss (all m copies received with error), it returns a NAK to the sender and starts a timer.
- The expiration of a timer also causes a retransmission request.
- Whenever the sender receives a NAK, it retransmits m data packet replicas.

Another protocol can be defined with the introduction of a mechanism that implements a NAK avoidance proposed by Ramakrishnan and Jain [17]. Whenever the receiver detects a packet loss it expects for a random time interval to return the NAK. In this system, the receiver multicasts its NAK to the other receivers. Whenever the other receivers realize that a NAK related to that packet has already been requested, they cancel its own request.

The **RMS** (**R**eceiver-initiated with **M**ultiple Copy and NAK **S**uppression) protocol has the following procedures:

- The sender multicasts m data-packet replicas to the receivers.
- Whenever a receiver detects a packet loss (all m copies with error), it expects during a random time before multicasting the NAK.
- During the random waiting time, if a receiver gets a NAK for the same packet from another receiver, it cancels its own NAK and start a timer, as it would have requested the retransmission.
- If the timer expires, it again causes the beginning of this process.
- Whenever a receiver sends a NAK, it starts a timer related to this NAK.
- Whenever the sender receives a NAK, m data-packet replicas are once more multicasted to receivers.

3 Performance Evaluation

The analysis evaluated hereby follows the same steps made in [18] and [16]. The model assumes that the multicast group is composed by R receivers and one sender. The new packets to be sent to the receivers are generated under a Poisson process with rate λ. The data-packet loss probability, p_r, is the same for all receivers. Further, these losses are assumed spatial and temporal independent. The acknowledgements are never lost. The propagation delay between the sender and each of the receivers has a value τ that is equal for all participants in the multicast system.

A random variable X is assigned to processing time of data packets. By the same way, the acknowledgments implie processing time that is another random variable, called Y. These random variables X and Y have means $E[X]$ and $E[Y]$ and second moments $E[X^2]$ and $E[Y^2]$, respectively. The processing of replicas is very simple and has usually little additional cost, because several functions like header processing are only once executed. Thus, its implementation becomes much easier and the processing is faster than using a more complex algorithm like a FEC coding. That is an advantage for this strategy and, due to this reason, the processing time of the m copies group is a random variable $X_g = X + f\sum_{i=1}^{m-1} X$, where f is a coefficient that means the reduction of the processing time for one replica, where $0 \leq f \leq 1$. Therefore, $E[X_g] = E[X] + f(m-1)E[X]$. Usually, this coefficient f will correspond to a low value.

Since the loss probability of one packet is p_r, the loss probability of the m-copies group will be $p = p_r^m$. It is useful to define another random variable $N_i^{(r)}$ that corresponds to the number of transmissions required for a correct reception of a data packet by the receiver i. This random variable has geometric distribution. Its probability density function may be written as

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

and its mean is $E[N^{(r)}] = \frac{1}{1-p}$.

By applying the same reasoning to all receivers, it is also useful to define the random variable N as the number of required transmissions for a correct reception by all receivers. Therefore, N is related to $N_i^{(r)}$ by the expression $N = max[N_i^{(r)}]$, $1 \leq i \leq R$. Its probability density function and its average are $P(N \leq n) = P(N^{(r)} \leq n)^R = (1-p^n)^R$ and $E[N] = \sum_{n=1}^{\infty}(1-(1-p^{n-1})^R)$.

3.1 Delay Analysis

These derived parameters are helpful in the subsequent delay analysis observed in **RM** and **RMS** protocols. The **RM** and **RMS** protocol delay consists on the elapsed time from the packet arrival at the sender buffer to its complete and correct reception.

3.1.1 Delay of Multicopy Protocol

In order to develop the **RM** protocol analysis, the waiting time in sender buffer must be obtained. The sender processes data and acknowledgements. Data will have an arrival rate λ_t^S. The acknowledgments are composed by not only NAKs that cause retransmissions but also NAKs that are processed but do not cause retransmissions due to the multiplicity of NAKs of the same packet that can arrive at the sender. Their arrival rates are $\lambda_r^S = \lambda(E[N]-1)$ and $\lambda_n^S = \lambda(RE[N^{(r)}]-E[N]-R+1)$, respectively.

The m data packets have processing time according to the random variable X_g. The NAKs that cause retransmission will have processing time $X_g + Y$ and the other NAKs simply Y. Thus, the total load at senders is $\rho^S = \lambda E[N]E[X_g] + \lambda R(E[N^{(r)}] - 1)E[Y]$.

As the data generation rate is assumed to be a Poisson process, the existing system at the buffer can be seen as a M/G/1 queue. Thereby, the mean waiting time at the sender can be obtained by the Pollaczec-Kinchin formula [23]:

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

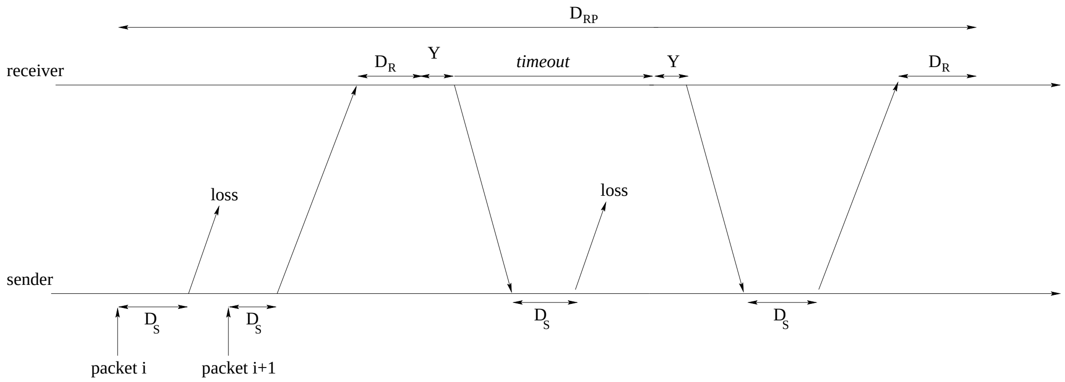


Figure 2: Possible operation of **RM** protocol.

In this system, two flows are present at the receivers. One is composed by received data and the other one by NAKs generated by the own receivers. Their service times are, respectively, X and Y. As the number of transmitted replicas are m and the distribution for received packets within a group is binomial, the mean number of received packets of a group is $m(1-p_r)$. Therefore, the arrival rates for data and NAKs are $\lambda_d^R = m\lambda E[N](1-p_r)$ and $\lambda_n^R = \lambda(E[N^{(r)}]-1)$, respectively, and the total load observed at receivers is $\rho_n^R = m\lambda E[N](1-p_r)E[X] + \lambda(E[N^{(r)}]-1)E[Y]$. Further, the mean waiting time at the buffers of receivers is:

$$E[W_{RM}^R] = \frac{\lambda_d^R E[X^2] + \lambda_n^R E[Y^2]}{2(1-\rho^R)} \ . \tag{3}$$

In order to obtain the delay, two phases can be distinguished. First, there is the time interval (random variable D_{RM}^{P1}) necessary to detect the loss and, later, another one (random variable D_{RM}^{P2}) needed to request the retransmission and wait for the complete reception [19]. Their means are:

$$\begin{aligned} E[D_{RM}^{P1}] &= \sum_{k=0}^{R-1} \begin{pmatrix} R-1 \\ k \end{pmatrix} p^k (1-p)^{R-k-1} \frac{p^{k+1}/\lambda}{1-p^{k+1}} \\ &\quad + E[W_{RM}^S] + E[W_{RM}^R] + \tau + E[X] + E[X_g] + E[Y] \end{aligned}$$

and $E[D_{RM}^{P2}] = \dfrac{p(T_R + E[W_{RM}^{Rn}] + E[Y])}{(1-p)} + E[W_{RM}^S] + E[W_{RM}^R] + E[X] + E[X_g] + 2\tau.$ (4)

For a receiver randomly chosen, a packet (any of the group) is not lost with probability $1-p$, and the delay is the sum of propagation delay, processing times and waiting times at both receiver and sender. Instead of that, the packet is lost with probability p and the delay is the sum of the elapsed periods of time on the two described phases. Finally, the total expected delay for this system can be obtained:

$$E[D_{RM}] = (1-p)(E[W_{RM}^S] + E[W_{RM}^R] + E[X] + E[X_g] + \tau) + p(E[D_{RM}^{P1}] + E[D_{RM}^{P2}]). \tag{5}$$

3.1.2 Delay of Multicopy Protocol with NAK Suppression

The observed delay of **RMS** protocol is found with similar analysis. Moreover, its operation and the one of **RM** protocol have identical elements, since both are receiver-initiated.

At the buffer of sender, there are once more 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 is less than with **RM** protocol due to NAK avoidance mechanism. This parameter is assigned to a random variable Q. Thus, the total NAK arrival rate not only at sender but also at other receivers is $\lambda E[Q]$, since NAKs are multicasted.

The arrival rates of data and the two kinds of NAKs are

$$\lambda_t^S = \lambda,\ \lambda_r^S = \lambda(E[N] - 1) \text{ and } \lambda_n^S = \lambda(E[Q] - E[N] + 1), \tag{6}$$

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

At present, the load and waiting time can be found by the relationships described in section 3.1.1 with the new arrival rates. The expression for $E[Q]$ is developed on [18].

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, can be easily found:

$$E[W_{RMS}^R] = \frac{\lambda_t^R E[X^2] + \lambda E[Q]E[Y^2]}{2(1 - \rho^R)}\ . \tag{7}$$

The **RMS** operation is also composed by the two phases identified in the **RM** protocol analysis. Nevertheless, the main difference between the two is the random time interval introduced in **RMS**, denoted by a random variable V. This interval is inserted between the phases. The expression for the expected value of V was derived in [18].

Finally, the average delay in a multicast system with **RMS** protocol is derived:

$$\begin{aligned} & E[D_{RMS}] \\ = \quad & E[W_{RMS}^S] + E[W_{RMS}^R] + E[X_g] + E[X] + \tau + pE[V] + \frac{p^2(T_R + E[W_{RMS}^{Rn}] + E[Y])}{1 - p} + \\ & +p\left(\sum_{k=0}^{R-1}\begin{pmatrix} R-1 \\ k \end{pmatrix} p^k(1-p)^{R-k-1}\frac{p^{k+1}/\lambda}{1 - p^{k+1}} + E[W_{RMS}^S] + E[W_{RMS}^R] + E[X_g] + E[X] + 2\tau\right). \end{aligned}$$

3.2 Processing Rate Analysis

The NAK implosion problem also creates a limitation for processing rate of the discussed systems. As the number of receivers grows the number of NAKs to be processed will get even higher and will increase the processing rate. Whenever delay gets higher, the achieved throughput gets lower. This section outlines a maximum throughput analysis of the described protocols.

3.2.1 Processing Rate of Multicopy Protocol

In order to find the processing rate achieved by the multicopy protocol, the defined **RM** protocol, the sender rate is first derived.

The total processing time is a random variable, called T^{RM}, that can be expressed as a sum of two processing times. 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

$$T_s^{RM} = \sum_{n=1}^{N} X_g + \sum_{i=1}^{L^{RM}} Y, \tag{8}$$

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

$$E[T_s^{RM}] = E[N]E[X_g] + E[L^{RM}]E[Y]. \tag{9}$$

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 every receiver is statistically equal, $E[L^{RM}]$ is simply $Rp/(1-p)$.

Therefore, the total processing time is expressed by

$$E[T_s^{RM}] = E[N]E[X_g] + E[L^{RM}]E[Y], \tag{10}$$

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. Once more for a correct 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 create the negative acknowledgements and return them to 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 that corresponds to necessary time to process 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:

$$1/\Lambda_r^{RM} = E[T_r^{RM}] = mE[N](1-p_r)E[X] + E[(N_r-1)^+]E[Y] + E[(N_r-2)^+]E[X_t]. \tag{11}$$

At last, the **RM** protocol throughput will be the minimum of the processing rate at sender and receivers:

$$\Lambda^{RM} = min\{\Lambda_s^{RM}, \Lambda_r^{RM}\}. \tag{12}$$

3.2.2 Processing rate of Multicopy Protocol with NAK suppression

The expressions for processing rates of protocol **RMS** are obtained in a similar fashion. The NAK suppression mechanism will cause a lower number of NAKs than the one 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 [16]. Without taking into account this situation, the expressions for processing rate at sender and receivers are, respectively,

$$1/\Lambda_s^{RMS} = E[T^{RMS}] = E[N]E[X_g] + (E[N]-1)E[Y] \text{ and} \tag{13}$$

$$1/\Lambda_r^{RMS} = E[T^{RMS}] = mE[N](1-p_r)E[X] + (E[N]-1)E[Y] + \frac{p^2}{1-p}E[X_t]. \tag{14}$$

By the same way, the throughput is equivalent to minimum of both processing rates:

$$\Lambda^{RMS} = min\{\Lambda_s^{RMS}, \Lambda_r^{RMS}\}. \quad (15)$$

4 Numerical Results

In this section, quantitative delay and processing rate results of **RM** and **RMS** protocols are showed.

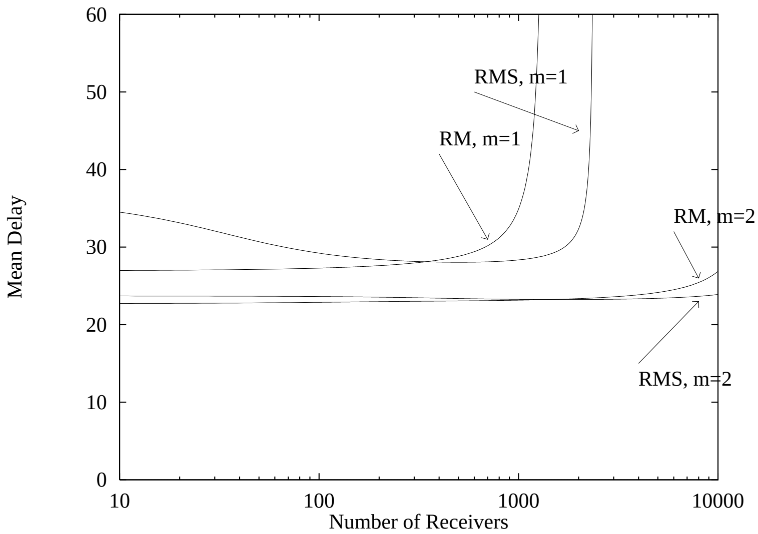


Figure 3: Delay versus number of receivers for $p = 0.1$ and $\lambda = 0.1$.

Kasera, Kurose and Towsley [9] have measured realistic values for these parameters with use of machines with processors Pentium 133MHz, operating system Linux, working in a networking environment with protocols UDP/IP and Ethernet. The processing times are 500 μs for data packets and 100 μs for ACK/NAKs. By normalizing these values, as made in [18], $E[X]$ assumes value 1.0 and $E[Y]$, 0.2. The propagation delay is 10 ms, thus its normalized value is $\tau = 20.0$. The loss probability is considered within a range observed in Mbone traffic [24], that is from 0.01 to 0.25.

Figure 3 shows the expected delay generated on the protocols taken into consideration, with loss probability $p_r = 0.1$ and data generation rate $\lambda = 0.1$. Another interesting fact is the reduction of number of acknowledgements by the NAK suppression mechanism and its implications on the provided scalability. The NAK suppression mechanism allows larger range of number of receivers with stability. The reduction of NAKs returned to the sender causes less processing for the participants of the multicast communications and, consequently, for a higher number of receivers, no NAK implosion will be experimented. This can be observed, by comparing the curves of **RMS** and **RM** protocols, for $m = 1$, and also for $m = 2$. **RM** and **RMS** protocols, for $m = 2$, work with almost unchanged values of their average delays up to 10,000 receivers. Furthermore, while the **RM** protocol for $m = 1$ has limitations from 1,000 receivers, for $m = 2$ the delay changes slightly. Only nearly 10,000 receivers the mean delay takes a positive variation. By comparing **RMS** protocol for $m = 2$ and $m = 1$, the same conclusions are taken. **RMS** for $m = 2$ even nearly 10,000 receivers does not present significant increase.

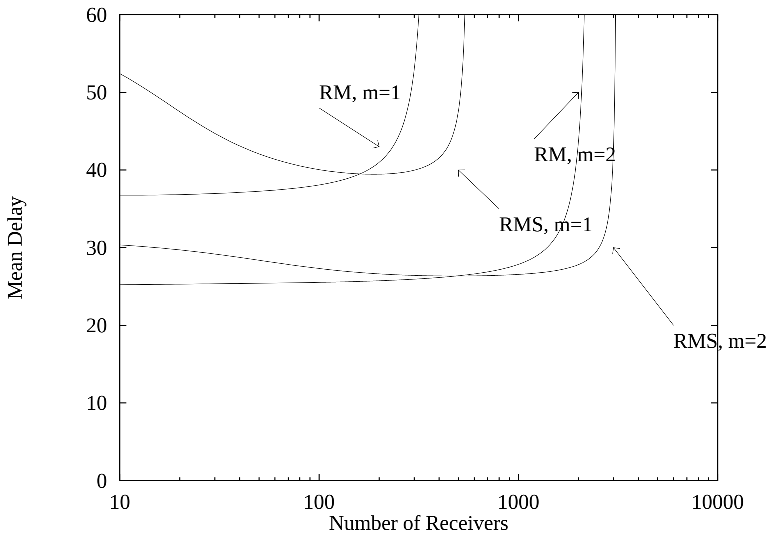


Figure 4: Delay versus number of receivers for $p = 0.25$ and $\lambda = 0.1$.

Another interesting observation that can be taken from the figure 3 is the decrease of the mean

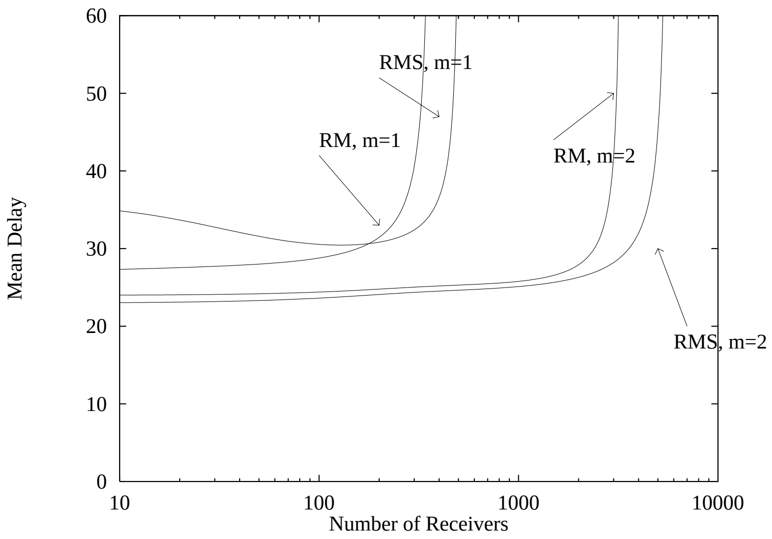


Figure 5: Average delay versus number of receivers for $p = 0.1$ and $\lambda = 0.2$.

delay for the **RMS** protocol on a range of low number of receivers. This fact happens due to NAK suppression mechanism. For a low number of receivers the random time interval to be expected will be as smaller as higher the number of receivers. Only after certain value of quantity of receivers, the influence of this parameter will get smaller, the processing will affect the performance and NAK implosion will cause limitations.

The curves shown on figure 4 allow observations for a loss probability $p = 0.25$. The same comparisons can be taken in a different degree, since the loss rate is higher. For instance, the limitation on number of receivers is larger. All curves present smaller ranges that work with stability than with $p = 0.1$. For **RM** and **RMS** protocols this limitation occurs nearly to 2,000 and 3,000 receivers, respectively. The decrease effect with raise of number of receivers due to NAK avoidance is once more noted and occurs strongly. As the loss rate is higher in this case, the random time interval will have stronger influence. The ratio between delay experimented by **RM** for $m = 1$ and $m = 2$ is another interesting fact. This ratio (around 1.48) shows the delay reduction obtained by the forward transmissions.

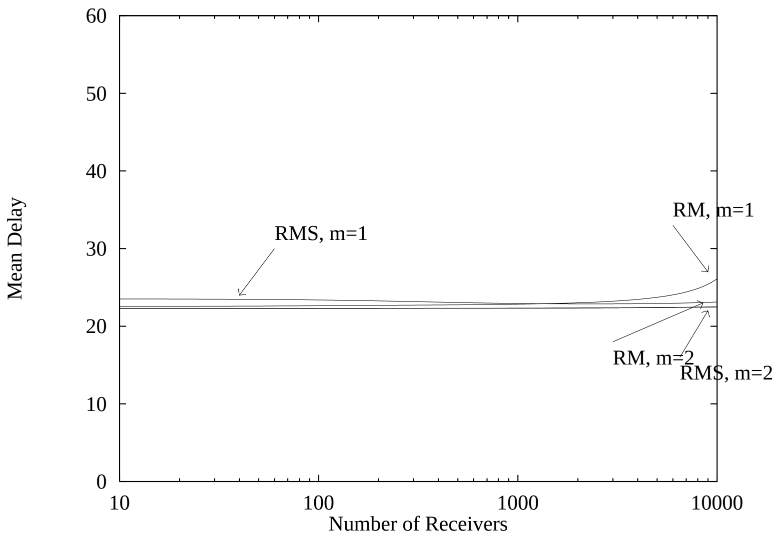


Figure 6: Average delay versus number of receivers for $p = 0.01$ and $\lambda = 0.1$.

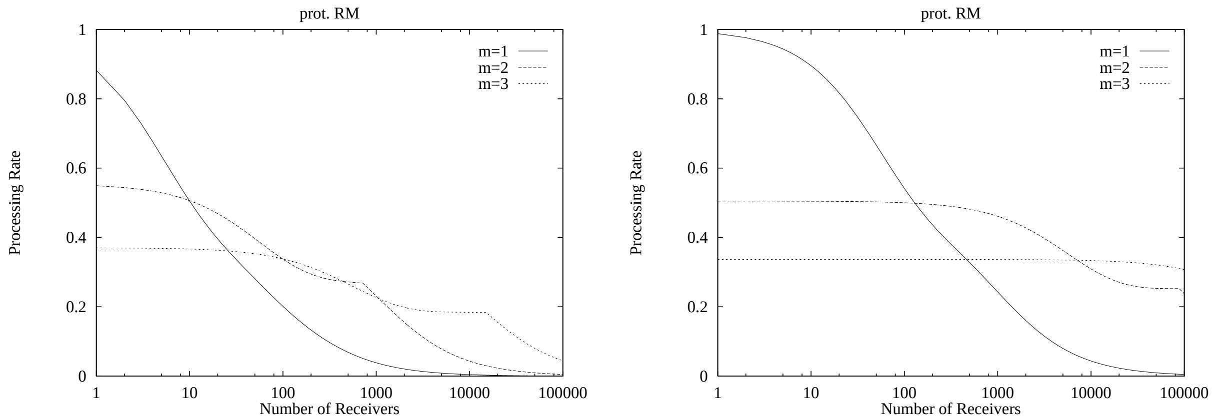
Figure 5 shows the average delay of the described protocols for loss probability $p = 0.1$ and data generation rate $\lambda = 0.2$, in order to understand the behavior of these systems. Beyond the previous comparisons, that are still valid, there are more strict limitations due to the increase in data generation rate. This is expected since the raise in number of packets is associated with increase of NAKs, and as a result a possible NAK implosion. The ratio discussed previously between delays of **RM** protocol for $m = 1$ 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 not critical influence under this situation.

At last, with lower loss rates, the pointed effects are less noticed. This can be observed in figure 6 as the curves are very close from each other. The described protocols have unchanged behavior on the whole considered range. Only near to 10,000 receivers the curve of **RM** protocol for $m = 1$ begins to show a stronger positive variation.

Following the analysis of processing rate, quantitative results can be drawn for the same values of processing time of packets and NAKs. The time values to process packets and NAKs have the same means as normalized values.

Figure 7(a) shows the throughput of **RM** protocol for integer values of m from 1 to 3, loss probability 0.1 and coefficient $f = 0.1$. The throughput achieved for $m = 1$ and a low number of receivers (approximately one receiver) is almost one. However, this throughput decreases quickly, when the number of receivers increases. In the range comprizing 1 to 10 receivers, the processing rate remains approximately constant around 0.5. From 1 receiver to 100 receivers, for $m = 3$, 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 7(a) show another important effect related to scalability. As for $m = 2$ and $m = 3$ the throughput decreases more slowly than for $m = 1$, the reached values for a high number of receivers with multiple copies are larger than with single packets. These effects can also be observed on figure 7(b), which presents the same curves for a value of probability of packet loss $p_r = 0.01$. In the curve for $m = 2$ the **RM** protocol presents an almost constant throughput in the range $1 - 1,000$ and a slow decrease of its value for higher number of receivers. For $m = 3$ and on the whole range $1 - 100,000$ the rate remains in a narrow band and the observed decrease of throughput is very slight.

Furthermore, the curves on figures 7(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 8(a) shows these rates for loss probability 0.1 and $m = 2$. Nearly 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 the packets that are copy of other ones take only a fraction of mean packet processing time, the sender rate is not decreased by a factor $1/m$.



(a) (b)

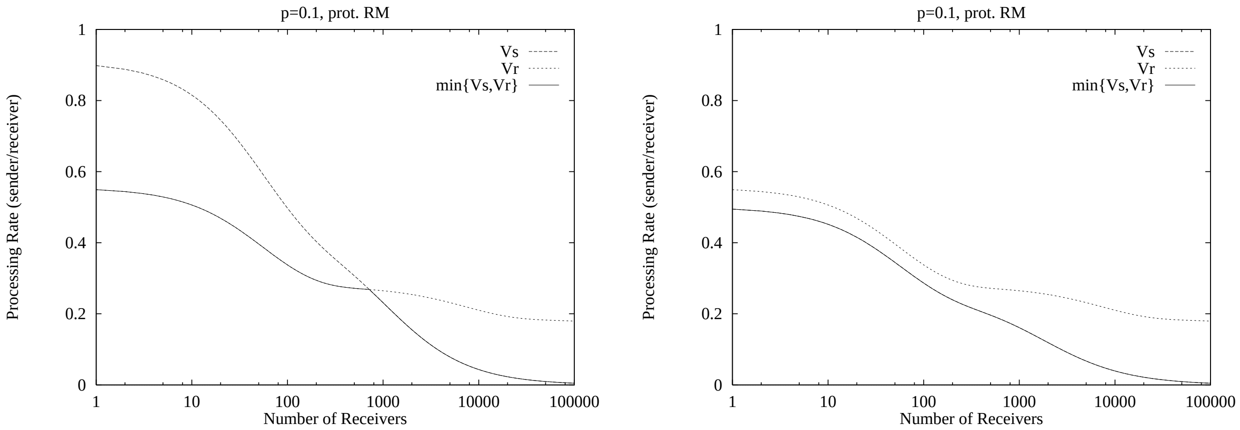
Figure 7: Processing rate for the **RM** protocol for $m = 1, 2, 3$ and: (a)$p = 0.1$, (b)$p = 0.01$.

Nevertheless, if a value one is assigned to coefficient f, this rate will certainly diminish by this factor. This is what takes place on figure 8(b), that was constructed with the same parameters but not the coefficient f, which is changed to value one.

Figure 9 presents **RMS** protocol processing rate, for $m = 1, 2$, and 3, loss probability 0.1 and coefficient $f = 0.1$. Once more for 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 to consider the coefficient $f = 1$. This is equivalent to consider that every copy will require the same mean time to process the packet at the senders. This can be noted on figure 10, that presents the three curves, for $p_r = 0.1$ as figure 7(a), but now with coefficient $f = 1$. The throughput for $m = 2$ decreases much slowly than for single copy packet. For $m = 3$ replicas the processing rate has a much slighter decrease.

Since the **RMS** protocol will cause closer values for throughput of systems with high number of receivers, but **RM** protocol not, it is also interesting to study the ratio between the throughput of both protocols. The following figures 11(a), 11(b), 11(c) show the ratio $\Lambda_{RM}/\Lambda_{RMS}$ for loss



(a) (b)

Figure 8: Processing rates for protocol **RM** at sender and receivers for: (a) $p = 0.1$ and $f = 0.1$, (b) $p = 0.1$ and $f = 1$.

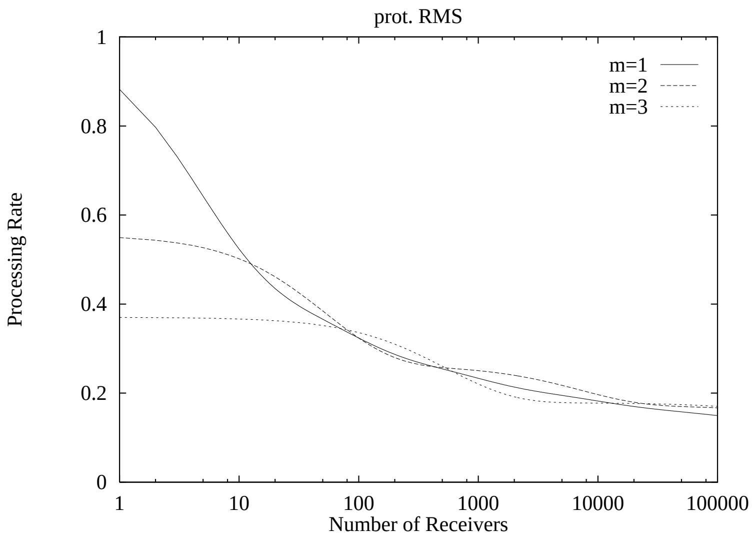


Figure 9: **RMS** protocol processing rate for $p = 0.1$ and $f = 0.1$, and $m = 1, 2, 3$.

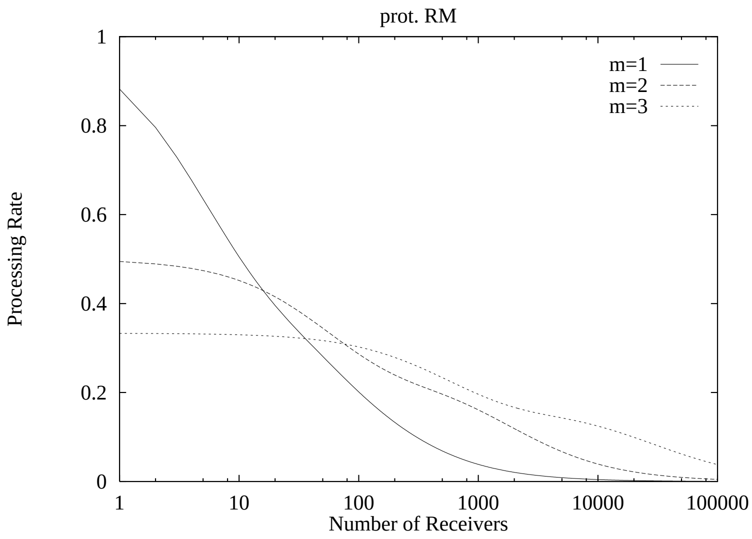


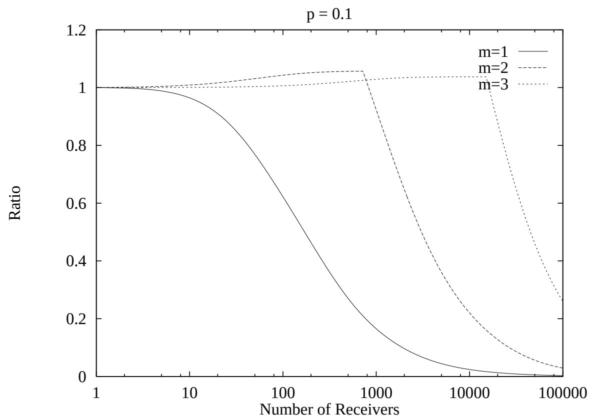
Figure 10: Processing Rate for **RM** protocol for $f = 1$, $p = 0.1$ and $m = 1, 2, 3$.

probabilities 0.1, 0.25 and 0.01, respectively. This ratio remains as closer to value one as m increases. On the other hand, as loss probability increases as smaller becomes the range in which there is ratio one. For $m = 3$ and $p_r = 0.01$, this ratio remains constant in the whole observed range.

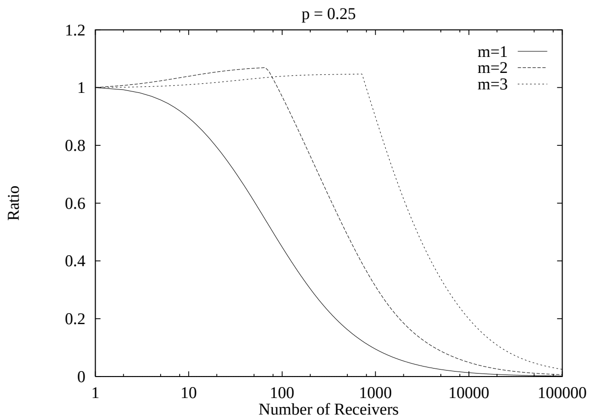
Finally, figure 12 shows the processing rate for several values of m with $p = 0.1$ and $f = 1$. With the increase of value of m, the throughput becomes so reduced and the difference between values $m = 4, 5$or 10 and $m = 3$, for instance, in range of high number of receivers is also small, that high values of m are not recommended. Thus, only $m = 2$ or $m = 3$ can be acceptable.

5 Conclusions

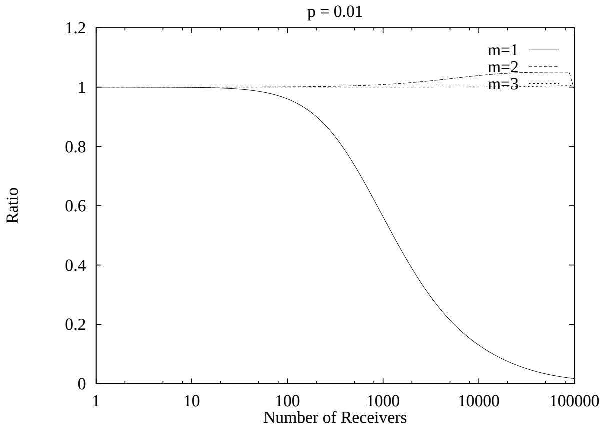
This paper has analyzed the delay and throughput achieved by using a multicopy scheme applied to reliable multicast communications. The multicopy reduces the number of retransmissions and, consequently, the NAK generation. Thus, it prevents from an acknowledgment implosion for a large number of receivers. Moreover, it reduces also the packet transmission delay, since a group of



(a)



(b)



(c)

Figure 11: Ratio $\Lambda_{RM}/\Lambda_{RMS}$ with same values of m and $f = 0.1$: (a)$p = 0.1$, (b)$p = 0.25$, (c)$p = 0.01$.

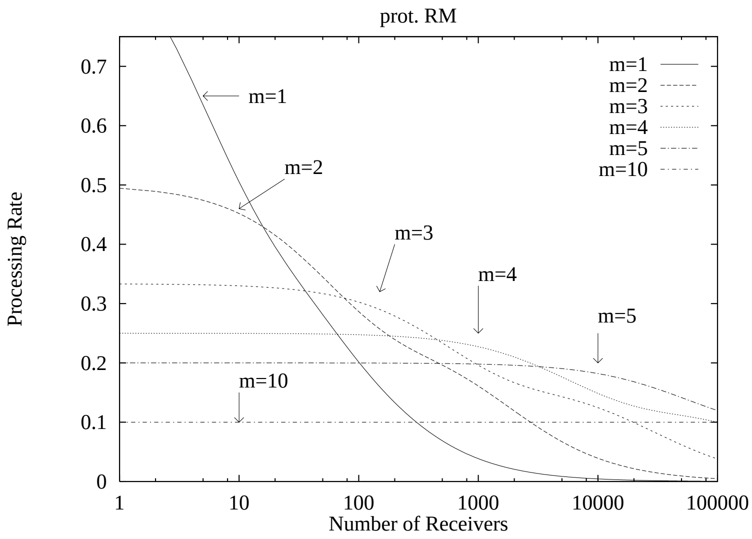


Figure 12: Processing rate of **RM** protocol for $p = 0.1$, $f = 1$ and $m = 1, 2, 3, 4, 5$ and 10.

packets is sent in advance.

The results have showed a stable behavior of delay for a larger number of receivers by using the multicopy than by a single copy scheme. The difference between achieved performance with and without the multicopy is great when the loss rates are high (loss probabilities $p = 0.1$ and 0.25). The obtained delay in a system with high data generation rate is also more well-behaved than when a single copy is used. The increase of data generation rate ($\lambda = 0.2$) in the obtained numerical results has also produced smaller delay for large number of receivers. The delay decreases when there is a low number of receivers (from 10 to 100) due to the transmissions in advance. The NAK suppression mechanism also improves the performance. However, as the delay remains stable from one receiver to 200 receivers under a single copy protocol, by using a multicopy the stable range increase to 1,000 receivers, instead of only 400 receivers by adopting a NAK supression mechanism.

The achieved throughput is approximately constant by using the multicopy in a large range of number of receivers. The achieved results for the multicopy with large number of receivers are greater than for the single copy. The decrease of the throughput is slow by using the multicopy.

However, by using many copies ($m = 4, 5, ...$) the throughput is very small and the difference between its values and the ones for $m = 2$ and 3 is little. Therefore, only $m = 2$ or 3 are recommended.

At last, the results have showed that the adopted scheme is robust for systems. Furthermore, changes that can possibly occur in conditions of communications such as loss rates and data generation rates affect the achieved performance less than without the strategy. Therefore, its adoption is extremely recommended for scalability purposes.

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, no. 2, 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, no. 2, Apr. 1998.

[14] 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.

[15] 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.

[16] 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, no. 3, Apr. 1997.

[17] 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.

[18] 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.

[19] 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*, Nov. 1998.

[20] A. R. K. Sastry, "Improving repeat-request (ARQ) performance on satellite channels under high error rate conditions", *IEEE Transactions on Communications*, vol. COM-23, no. 4, pp. 436–439, Apr. 1975.

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

[22] 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.

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

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