A PRIORITY SCHEME APPLIED TO SCALABLE RELIABLE MULTICAST COMMUNICATIONS

Daniel Antunes Maciel Villela
daniel@gta.ufrj.br

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

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

Abstract

Scalability in reliable multicast communications is a primal requirement, since there is a large growth of distributed applications based on internetworking with global dimensions. This paper proposes a priority scheme applied to three reliable multicast protocols, resulting in different generic models and performance evaluation for these models is established. The results show that the priority scheme ensures more robustness when possible variations of network states take place. There is a slight difference for the mean delay experimented under this scheme, when an increase in loss probability or in network load occurs. The mean delay has also little sensibility due to raise of number of participants in the multicast system.

Keywords: Reliable Multicast Protocols, Scalability and Performance Evaluation.

1 Introduction

Scalability in multicast communications is essential, since many applications are being developed and the number of participants is growing in a higher rate. The MBone is at such a platform and contains a number of these applications like whiteboard, for instance. High speed networks and multimedia need increase the distributed application demand. Thus, there is a set of requirements for a class of transport protocols that operate in these systems. They must be reliable, efficient, and scalable under variation of number of participants.

The requirement of reliability implies acknowledgments that are returned to the sender. These acknowledgments can be positive (ACK) or negative (NAK). When the communication groups increase, the number of acknowledgments may become very high and cause an overload. This effect is called acknowledgment implosion and can lead to overhead in sender processing and lack of resources needed by the traffic. Therefore there are difficulties to maintain the quality of service (QoS) required by applications.

Some solutions for this problem try to diminish the number of acknowledgments and reduce the effects related to implosion problem. Some authors have proposed hierarchies as tree and ring topologies [1, 2] and local groups [3]. Biersack *et al.* proposed the utilization of Forward Error Control (FEC) with the Automatic Repeat Request (ARQ) in an ARQ hybrid scheme [4]. De Lima and Duarte [5] have analyzed an ARQ strategy with selective repeat for multicast communications. Floyd *et al.* [6] introduced the Application Level Framing (ALF) concept. Pingali, Kurose and Towsley [7] made an analysis for reliable multicast protocols models and a comparison of sender-initiated protocols, receiver-initiated protocols and NAKs suppression mechanism [8, 6]. Their study is based on a throughput protocol analysis. Yamamoto *et al.* [9] used these models for delay analysis.

This paper presents an analysis related to delay occurred in reliable multicast protocols. A priority scheme is proposed in order to reduce the effects caused by network state variations like participant number changes. The priority mechanism and general paradigms are used to define three generic protocols, that are based on the models presented on [7, 9] for a comparison between the analyzed systems with other ones that work without priorities. The results show efficiency and robustness in systems under the defined protocols in terms of scalability and change of network parameter values.

This paper is structured as follows: the section 2 describes the scheme proposal and the three models that adopted it. In section 3 the performance evaluation of these models is performed. Some results are presented in section 4. Finally, the section 5 has final comments and the future work purposes.

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

2 Reliable Multicast Protocols with Priority Scheme

This paper introduces a priority scheme to be used in reliable multicast communications and defines three generic protocols that adopted it and operate with well-known acknowledgement mechanisms presented on previous works [8, 6, 7]. Their operation is briefly discussed in this section.

2.1 Service Discipline with a Priority Scheme

For two flows, a priority scheme can grant more privilege to one flow instead of the other. These flows are referred to as the priority classes. Whenever there is work to do in both two classes, the handling routine will serve primarily the highest priority class. At each class separately the service ordering is done by the simple FCFS (First Come First Served) discipline. Thereby, when there is only one present flow, the service occurs by temporal ordering. Moreover, the priority scheme adopted hereby is non-preemptive. Thus, if the lowest priority flow is being served and a packet of the highest priority flow arrives, the service is completely done before the new service beginning. These flows can be observed as a queuing model, where there are queues (flows) served by a unique server and each one have an assigned priority. The figure 1 shows this model for two flows, one composed by data packets and the other one by acknowledgment packets.

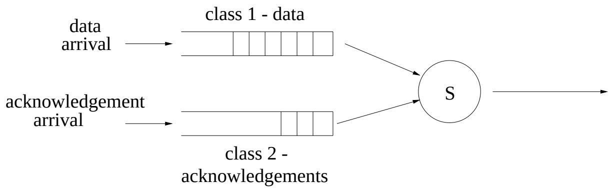

Figure 1: The priority scheme seen as two queues.

Without priorities, the participants of multicast communication system deal identically with the existing flows. For NAK-based protocols, the raise of acknowledgement number causes a load bigger than there was before at the senders. This can happen due to a raise of packet loss rate, for instance. However, the flow exclusively generated by data remains with the same load. As the processing applied to each flow is exactly the same, a performance loss for the data flow will take place. Some important measures like delay can change significantly. The same effect can be predicted in consequence of an increase in number of receivers. Therefore, this scenario is highly undesirable for scalability.

A priority scheme, where the highest priority flow is assigned to the data flow, enables more insensible handling within the possibility of changes in system behavior. To serve the data flow as well as in suitable conditions is the objective, because it will allow some quality of service guarantees.

2.2 Protocols with the Priority Scheme

By using this priority scheme, a first protocol can be defined. This one uses positive acknowledgement. According to Pingali, Kurose and Towsley [9, 7], it can be classified as a sender-initiated protocol. The sender multicasts packets to receivers and immediately starts a timer. After a correct receipt, a receiver replies to the sender with an ACK. Whenever occurs the expiration of the timer (timeout), the sender repeats its transmission. The sender treats the received ACKs whenever there are no data packet. Otherwise, it processes entirely the data flow, before the acknowledgement treatment. This protocol will be called **SP** (Sender-initiated with Priorities).

Instead of sender-initiated protocols, another protocol is defined with the use of negative acknowledgement and the priority scheme applied. The sender once more multicasts data packets to receivers. A receiver that detects a packet loss will ask for its retransmission with a negative acknowledgement (NAK) and automatically will start a timer. The expiration of a timer produces a new retransmission request. A NAK received is treated if there is no packet data in buffer waiting to be sent. Otherwise, it expects the end of data packet transmission. This protocol will be named **RP** (Receiver-initiated with Priorities).

A third protocol, **RPS** (Receiver-initiated with Priorities and NAK Suppression), can be defined with the introduction of a mechanism that implements a NAK avoidance proposed by Ramakrishnan and Jain [8]. 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. Whenever the other receivers realize that a NAK related to that packet has already been requested, they cancel its own request. Thus, the returned NAKs will be multicasted to the sender and all receivers.

It can be seen that the adopted retransmission fashion in the three defined protocols is selective repeat. These protocols are similar to the ones pointed in [7], which makes use of sender-initiating and receiver-initiating concepts and NAK suppression mechanism, but makes no difference between the flows.

3 Performance Evaluation

The delay generated in systems that work with the **SP**, **RP**, and **RPS** protocols consists on the elapsed time since the packet arrival at the sender buffer until its correct and complete reception.

The analysis evaluated hereby has some assumptions and follows the same steps made in [9]. 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 is the same for all receivers. Further, these losses are assumed independent spatially and temporally. 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 implies 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.

It is useful to define another random variable $M_i^{(r)}$ that corresponds to the number of transmissions required for a correct receipt of a data packet by the receiver i. This random variable has geometric distribution. Its probability density function may be written as $P(M^{(r)} = m) = (1-p)p^{m-1}$, $m \geq 1$, and its mean is $E[M^{(r)}] = \frac{1}{1-p}$.

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

These parameters are helpful in the subsequent analysis, in which the delay observed in systems with **SP**, **RP**, and **RPS** protocols is studied.

3.1 Delay of SP Protocol

In order to obtain the observed delay in a system under the **SP** protocol, it is necessary to calculate the waiting times of the data packets in buffers. At the sender two flows are noticed. There is one flow that corresponds to data packets to be multicasted to receivers for the first time or in a possible retransmission. The generation rate of new data packets is λ. The rate of retransmitted packets is $\lambda(E[M]-1)$. Thus, the total rate for data flow is $\lambda_d^S = \lambda E[M]$. As packets have processing time X, the sender data load is $\rho_d^S = \lambda E[M]E[X]$.

Similarly, there is one acknowledgment flow. Its arrival rate will be $\lambda_a^S = \lambda R E[M](1-p)$. The load caused by this flow can be written as $\rho_a^S = \lambda R E[M](1-p)E[Y]$.

At the sender the priority scheme is applied since there are two distinct flows. The arrival processes are assumed Poisson and service times are independent random variables. By using that, this system is modeled by a M/G/1 queue with priorities. Thereby, the waiting time of data packets in buffer is

$$E[W_{SP}^{Sd}] = \frac{\lambda_d^S E[X^2] + \lambda_a^S E[Y^2]}{2(1-\rho_d^S)} \,. \tag{1}$$

After computing waiting times at senders, to obtain waiting times at receivers is the following necessary step. In this situation there is only one flow composed by data packets. Its arrival rate is $\lambda^R = \lambda E[M](1-p)$. Since a correct receipt of a data packet is followed by an ACK sending, processing time is $X+Y$. Thus, the load at the receiver is $\rho^R = \lambda E[M](1-p)(E[X]+E[Y])$. The waiting time is found by the Pollaczec-Kinchin formula:

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

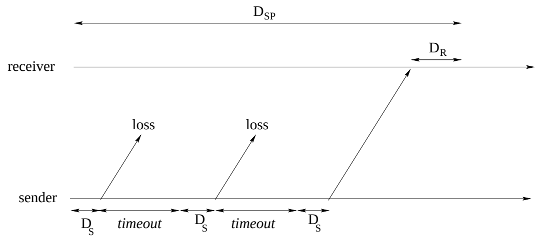

Figure 2: Possible operation of system under **SP** protocol.

Finally, the period of time elapsed between the packet arrival at sender buffer and its complete reception by the receiver is denoted D_{SP} (figure 2). As the probability of j transmissions for a given receiver is $p^{(j-1)}(1-p)$ and the timeout at sender is a constant value T_S, the expectation of D_{SP} is given by the expression

$$E[D_{SP}] = \frac{(E[W_{SP}^{Sd}] + E[X] + pT_S)}{(1-p)} + \tau + E[W_{SP}^R] + E[X]. \tag{3}$$

3.2 Delay of RP Protocol

The **RP** protocol analysis follows the same reasoning made for **SP**. First, the waiting time in buffer must be obtained. The sender once more processes two flows, one containing data and other one consisting on acknowledgements. The first one has arrival rate λ_t^S. The second one has not only NAKs that cause retransmissions but also NAKs that are processed but do not cause it. Their arrival rates are $\lambda_r^S = \lambda(E[M]-1)$ and $\lambda_n^S = \lambda(E[M^{(r)}] - E[M] - R + 1)$, respectively:

The data packets have processing time according to the random variable X. The NAKs that cause retransmission will have processing time $X+Y$ and the other NAKs simply Y. Thus, the total loads related to data packets and NAKs are $\lambda_r^S = \lambda(E[M]-1)$ and $\lambda_n^S = \lambda(E[M^{(r)}] - E[M] - R + 1)$.

As the data flow has higher priority than the acknowledgment flow, the mean waiting time of data packets and acknowledgments can be obtained:

$$E[W_{RP}^{Sd}] = \frac{\lambda_t^S E[X^2]+\lambda_r^S(E[X^2]+E[Y^2]+2E[X]E[Y])+\lambda_n^S E[Y^2]}{2(1-\rho_d^S)}$$

and $E[W_{RP}^{Sn}]$

$$= \frac{\lambda_t^S E[X^2]+\lambda_r^S(E[X^2]+E[Y^2]+2E[X]E[Y])+\lambda_n^S E[Y^2]}{2(1-\rho_d^S)(1-\rho^S)},$$

where $\rho^S = \rho_d^S + \rho_n^S$.

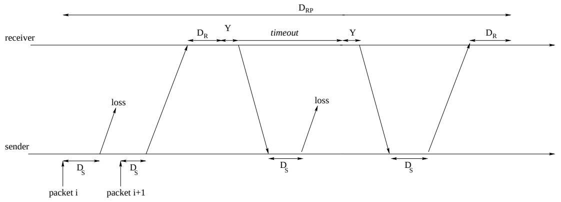

Figure 3: Possible operation of system under **RP** protocol.

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. Therefore, the arrival rate and load of data packets are $\lambda_d^R = \lambda E[M](1-p)$ and $\rho_d^R = \lambda E[M](1-p)E[X]$, and the same parameters from NAKs, $\lambda_n^R = \lambda(E[M^{(r)}]-1)$ and $\rho_n^R = \lambda(E[M^{(r)}]-1)E[Y]$. Further, the mean waiting times for the recognized flows are:

$$E[W_{RP}^{Rd}] = \frac{\lambda_d^R E[X^2] + \lambda_n^R E[Y^2]}{2(1-\rho_d^R)} \text{ and} \tag{4}$$

$$E[W_{RP}^{Rn}] = \frac{\lambda_d^R E[X^2] + \lambda_n^R E[Y^2]}{2(1-\rho_d^R)(1-\rho^R)} . \tag{5}$$

In order to obtain the delay, two phases can be distinguished. First, there is a time interval (random variable D_{RP}^{P1}) necessary to detect the loss and, secondly, another one (random variable D_{RP}^{P2}) needed to request the retransmission and wait until the complete receipt. Their means are:

$$\begin{aligned} E[D_{RP}^{P1}] &= \sum_{k=0}^{R-1} \binom{R-1}{k} p^k (1-p)^{R-k-1} \frac{p^{k+1}/\lambda}{1-p^{k+1}} \\ &\quad + E[W_{RP}^{Sd}] + E[W_{RP}^{Rd}] + \tau + 2E[X] + E[Y] \end{aligned}$$

and $E[D_{RP}^{P2}]$

$$\begin{aligned} &= \frac{p(T_R + E[W_{RP}^{Rn}] + E[Y])}{(1-p)} + E[W_{RP}^{Sd}] + \\ &\quad + E[W_{RP}^{Rd}] + 2(E[X]+\tau). \end{aligned}$$

For a receiver randomly chosen, the packet 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 now the sum of the periods of time elapsed on the two described phases. Finally, the total expected delay for this system can be obtained:

$$\begin{aligned} E[D_{RP}] &= (1-p)(E[W_{RP}^{Sd}] + E[W_{RP}^{Rd}] + 2E[X] + \tau) + \\ &\quad p(E[D_{RP}^{P1}] + E[D_{RP}^{P2}]). \end{aligned} \tag{6}$$

3.3 Delay of RPS Protocol

The observed delay in a system under **RPS** protocol is found with similar analysis. Moreover, its operation and the one of RP protocol have identical parts, since both are receiver-initiated.

At the sender, there are once more flows composed by data and acknowledgments. The acknowledgment flow is 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 **RP** 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 $\lambda_t^S = \lambda$, $\lambda_r^S = \lambda(E[M]-1)$ and $\lambda_n^S = \lambda(E[Q]-E[M]+1)$, with service times X, $X+Y$ and Y, respectively.

At present, the load and waiting time can be found by the relationships described in section 3.2. The expression for $E[Q]$ is developed on [9].

At receivers, the data packet flow has arrival rate $\lambda_t^R = E[M](1-p)$ and the load $\rho_d^R = \lambda E[M](1-p)E[X]$. The arrival rate of NAKs is already known and the load of this flow is $\rho_n^R = \lambda E[Q]E[Y]$.

Consequently, the mean waiting times of data packets, W_{RPS}^{Rd}, and acknowledgment, W_{RPS}^{Rn}, can be found:

$$E[W_{RPS}^{Rd}] = \frac{\lambda_t E[X^2] + \lambda E[Q]E[Y^2]}{2(1-\rho_d^R)} \text{ and} \tag{7}$$

$$E[W_{RPS}^{Rn}] = \frac{\lambda_t E[X^2] + \lambda E[Q]E[Y^2]}{2(1-\rho_d^R)(1-\rho^R)}. \tag{8}$$

The **RPS** operation is also composed by the two phases identified in the **RP** protocol analysis. Nevertheless, the main difference between the two is the random time interval introduced in **RPS**, denoted by a random variable V. This interval is inserted between the phases. The expression for the expected value of V was derived in [9].

Finally, the average delay in a multicast system with protocol **RPS** is found following the same reasoning of equation 6 with the expressions derived above.

4 Quantitative Results

In this section, some quantitative results obtained from the analysis of **SP**, **RP** and **RPS** protocols are showed and compared with similar ones that do not make any distinction between flows. These ones are **A**, **N1** and **N2** defined in [7, 9]. The effects generated by changes of number of participants are measured by making variations on number of receivers.

The processing times are considered uniformly distributed. Kasera, Kurose and Towsley [3] have measured realistic values for these parameters with use of

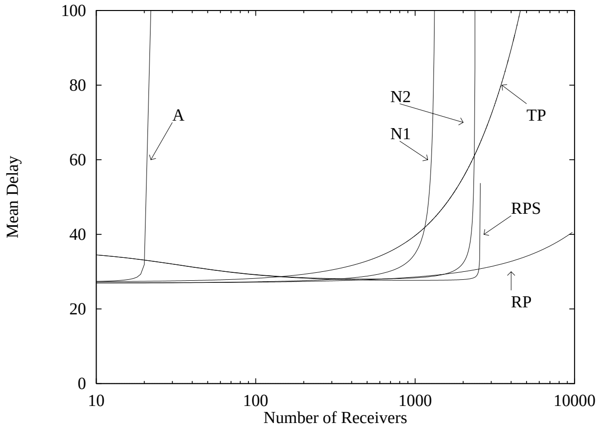

Figure 4: Average delay versus number of receivers with $p = 0.1$ and $\lambda = 0.1$.

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 [9], $E[X]$ assumes value 1.0 and $E[Y]$, 0.2. The propagation delay is 10 ms, thus its normalized value is $\tau = 20.0$.

The curves on figure 4 show the expected delay generated on systems under all protocols taken into consideration, with loss probability $p = 0.1$ and data generation rate $\lambda = 0.1$. By looking at curves of **SP** and **A**, it is not difficult to see that the delay of **A** increases with a high rate, when number of receivers increases, whereas the delay of protocol **SP** also presents an increase although smoother than **A**. Further, in figure 4, **RP** shows the same behavior, by comparing it with **N1** protocol.

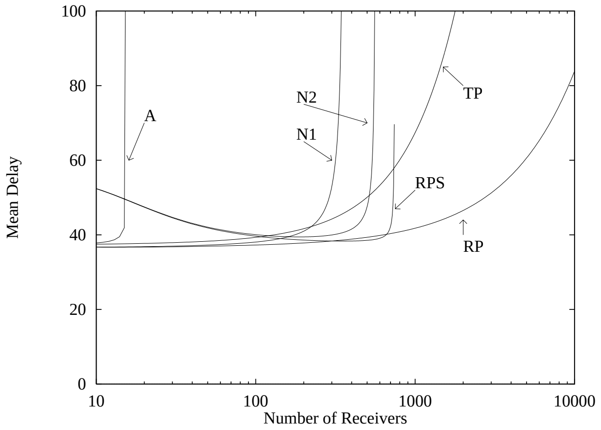

Figure 5: Average delay versus number of receivers with $p = 0.25$ and $\lambda = 0.1$.

The same observations can be made for the comparison between protocols **RPS** and **N2**. There is a different characteristic for these curves. A slight reduction of delay occurs in range from 10 to 100 receivers due to NAKs avoidance mechanism. This effect happens because the waited random time interval is more probably smaller for high number of receivers than for a low number. Thus, in the pointed range, these amounts contribute more significantly. Nevertheless, the observed variation in this range is not large. For large number of receivers, the protocol **RPS** also presents increasing values. This happens, because the acknowledgement load becomes extremely high. The waiting time of NAKs in buffer is part of the generated delay, since time necessary for the first NAK to be sent must be computed. Therefore, this raising time explains the noted effect. A comparison between sender-initiated and receiver-initiated protocols shows that the expected delay for the former is bigger than the latter. This is already expected, since the loss probability used causes a higher number of positive acknowledgments than negative ones. Another interesting fact is the reduction of number of acknowledgements by the NAK suppression mechanism and its implications on provided scalability.

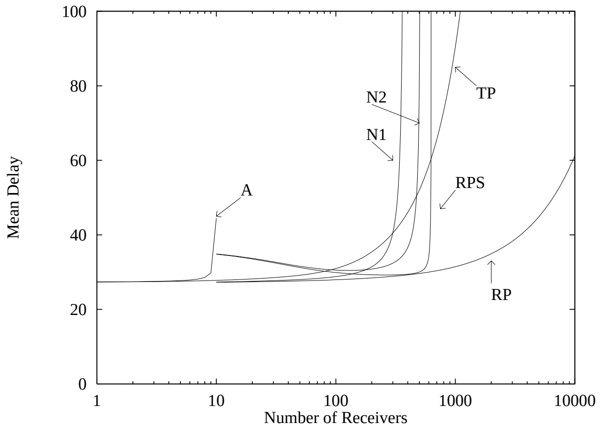

Figure 6: Average delay versus number of receivers with $p = 0.1$ and $\lambda = 0.2$.

The curves shown on figure 5 allow to establish the comparison for a loss probability ($p = 0.25$). Once more, the priority scheme used in **SP**, **RP**, and **RPS** allows little variation if they are compared with **A**, **N1**, and **N2**. Scalability is higher for little loss probabilities than for large ones, since acknowledgement information is relatively little. Furthermore, there is a slight variation on expected delay with loss probability changing. This effect is also noted on figure 7 where a loss probability $p = 0.01$ is used, even though it happens in a much lower degree.

There is a difference between the delay for set **SP**,

RP, and **RPS** and the set **A**, **N1** and **N2**, for higher data generation rates. The explanation for this fact also comes from the increase of number of acknowledgements. The figure 6 shows the behavior of the systems operating under the six described protocols at a arrival rate $\lambda = 0.2$.

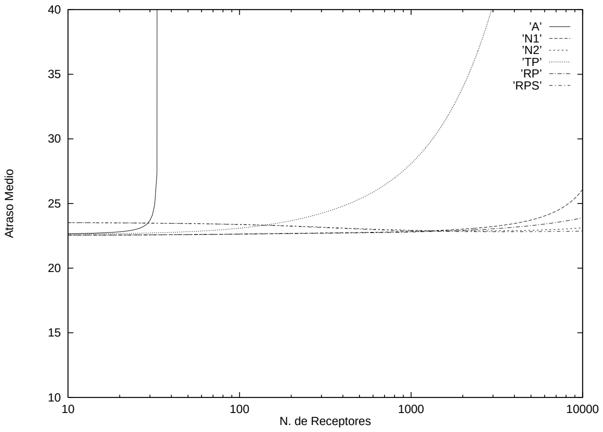

Figure 7: Average delay versus number of receivers with $p = 0.01$ and $\lambda = 0.1$.

5 Conclusion

This paper presented the introduction of a priority scheme applied to reliable multicast communications. The main objectives are scalability and high performance. Three generic protocols were defined with the proposed scheme, the concepts of sender-initiating and receiver-initiating several approaches. One of them was the adoption of sender-initiating and receiver-initiating concepts. Another used possibility was the NAK suppression mechanism.

Performance evaluation of each defined protocol was developed. The assumed model enables general distribution for processing times of data and acknowledgment. The relationship for the expected delay in each situation was found. A comparison between protocols with and without priorities was achieved.

The results show that the adopted scheme allows more robustness, because low variation of average delay is caused by the changes of state in communication channel, including loss probability and data generation rate alterations. In terms of scalability, the variation is also slight when there is a raise of number of receivers in systems. The increase rate of delay is smoother with priority than without the scheme. Therefore, there is a large degree of insensibility due to alterations in number of participants of the multicast system.

The future works purposes include studies of models with more generality. Variations of these models are also interesting in order to increase performance and scalability and to allow adaptive mechanisms.

References

[1] 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.

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

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.