Efficient Loss Recovery Algorithm for Multicast Communications in Active Networks

Marta Barría[1] and Reinaldo Vallejos[2]

[1]Computer Science Department, University of Valparaíso, Chile
marta.barria@uv.cl

[2]Department of Electronics Engineering, Technical University Federico Santa María, Chile
reinaldo@elo.utfsm.cl

Abstract.- A novel highly scalable loss recovery algorithm for Multicast transmissions in active networks that achieves near-optimal implosion and very low latency is proposed. Quasi-minimum implosion is attained by stochastically selecting a sub-set of the loss-affected members as NACK senders. If appropriately tuned, the algorithm selects with high probability only one member as a NACK sender. Near-optimal latency is obtained by minimizing the time taken to select the NACK sender and by retransmitting from the closest possible location. Performance evaluation results show a maximum implosion 4% higher than the optimal value and a low latency.

Index Terms. Multicast Communication, Loss Recovery, Reliability, Active Networks, Multicast Tree.

1 Introduction

Multicast communication offers an efficient way to disseminate information from one or more transmitters to a group of receivers. Reliable Multicast applications require that each destination receive correctly all transmitted packets. Examples of this type of application are software distribution, shared whiteboards, interactive games, network banking and replication of databases [2, 3]. Although there exist lower-layer technologies for providing reliable transmission, the best-effort nature of Internet Protocol hampers error-free packet reception. Reliable multicast requires a scalable recovery of losses. The two main impediments to scale are implosion and recovery latency. Implosion occurs when the loss of a packet trig-

Please use the following format when citing this chapter:

Barria, M. and Vallejos, R., 2007, in IFIP International Federation for Information Processing, Volume 229, Network Control and Engineering for QoS, Security, and Mobility, IV, ed. Gaïti, D., (Boston: Springer), pp. 231–241.

gers simultaneous redundant requests and/or retransmissions from many receivers [3]. The techniques employed to decrease implosion may introduce a long latency recovery. The ideal situation is that the size of the implosion be equal to one and the latency be minimum. For this reason the error recovery mechanisms that have been proposed in specialized literature seek to reduce implosion and latency simultaneously. However, since both objectives are conflictive, algorithms attempt to obtain a compromise between them.

Among the different approaches for loss recovery in Multicast communications, receiver-initiated schemes (receivers detecting a loss send a negative acknowledgement (NACK)) have been shown to perform better than positive acknowledgements (ACK) schemes [11]. One approach to improving reliability is through the use of receiver-initiated protocols with local recovery [1, 3, 4, 8, 9]. A further technique for improving Multicast reliability has emerged from the active networks area [7]. In active networking the routers themselves play an active role by executing application-dependent services on incoming packets. The advantages of using this architecture with reliable multicast protocols are: the cache of data packets allows for local recoveries of loss packets and reduces recovery latency, the global or local suppression of NACKs reduces the NACK implosion problem, and the partial multicast of repair packets to a set of receivers limits both retransmission scope and bandwidth usage. ARM [6], AER[5] and DyRAM [7] are examples of newly proposed reliable multicast protocols that use active networks.

In this article we present a new local recovery algorithm for Multicast communications based on active routers that achieves near-minimum implosion and latency simultaneously.

The rest of this paper is organized as follows: in Section 2 the new error-recovery algorithm is presented; Section 3 contains the performance evaluation of the algorithm; Section 4 provides and discusses numerical examples; and finally, in Section 5 the conclusions are presented.

2 Proposed Algorithm

2.1 Notation

The network is modeled by the graph $G=(V,E)$, where V corresponds to the set of all nodes in the network and E to the set of all its links. The network and the Multicast group are composed of normal routers, active

routers and receivers. The routers form the set L, and the hierarchy between them is created using a distribution tree D generated by a Multicast routing protocol [12]. The information is originated in the source, denoted s. The receivers form the set $R = \{r,\ 1 \le r \le |R|\}$. Hence, all the members of the Multicast group together form the set $M = L \cup R \cup \{s\}$.

2.2 Algorithm

Under normal operation, packets originated at the source are transmitted to all receivers through the distribution tree D. Under error conditions, the proposed algorithm, called Loss Recovery Algorithm for Reliable Multicast (LRARM), is activated. LRARM's operation can be described in the following eight points:

I. If an active router $m \in L$ detects a loss:
   1. It forwards an inhibit message to each of its receivers and its child routers in D;
   2. It executes a Bernoulli random experiment with parameter $p_1(m)$ (to be described below). Upon successful outcome of the random experiment, it sends a NACK up to its parent router in D.
   3. It starts a timer, with timeout $TO(m)$ (described below).
   4. It continues the normal transmission of another received packets.
   5. If $TO(m)$ expires and the requested packet has not been received, m repeats steps 1. through 5. However, the parameter of the Bernoulli experiment now changes to $p_n(m)$ (described below), where n denotes the nth execution of this step (i.e., step 5.) for the unreceived packet.

II. In the case where a receiver $m \in R$ detects a loss, it executes the steps I.1 through I.5 as described above.

III. When an active router receives the inhibit message:
   1. It forwards the message downstream to each of its receivers and its child routers in D.
   2. It refrains from sending a NACK upstream for this packet.

IV. When a receiver receives the inhibit message, it refrains from sending a NACK for this packet.

V. When an active router receives a NACK:
   1. It retransmits the requested packet to its child routers in D.
   2. It eliminates the NACK from the network.

VI. When a receiver receives a NACK, it eliminates the NACK from the network.

VII. When an active router receives the retransmission of a lost packet:
   1. If the packet has not been received before, it forwards the packet to its receivers and its child routers;
   2. Otherwise, it discards the packet.

VIII. When a receiver receives the retransmission of a lost packet that it has received before, that packet is discarded.

Point I is essential for achieving near-minimum implosion by reducing the number of NACK senders (using the inhibit message). The parameter of the Bernoulli distribution is chosen in such a manner that there is a high probability only one potential NACK sender will be selected to transmit the NACK. In addition, low latency is assured by quickly selecting the NACK sender(s) (a process that takes a few microseconds, the time necessary for a CPU to perform the Bernoulli experiment) and by maintaining a copy of transmitted packets in every active router associated with the Multicast group (only those packets with unexpired timeouts are maintained; as timeouts expire without receiving NACKs, the corresponding packets are discarded). Therefore, the closest parent active router to the point of failure is responsible for the retransmission.

2.3 Timeout Values.

Assume that a packet loss has occurred in link $e_{j,k}$ of D. The recovery tree R_k is then defined as the subtree of D, made up of: the first parent active router of k, named $\rho(k)$; the set of active routers of the multicast group that are the first descendents of $\rho(k)$ in D; the receivers of the multicast group, and all the paths that interconnect them. Each recovery tree has its own timeout, calculated upon connection set-up (and when a receiver joins or leaves the group) and known by all its active routers and receivers.

The timeout of R_k, $TO(R_k)$, corresponds to the round trip time between $\rho(k)$ and its farthest member (active router or receiver). Given a leaf $m \in R_k$, $TO(R_k)$ can also be denoted as $TO(m)$.

2.4 Bernoulli Parameter Values.

The value of $p_n(m)$, determined upon connection set-up (and when a receiver joins or leaves the group), is given by:

$$p_n(m) = \begin{cases} \min\left(1; \dfrac{l(m)}{\sum_{\forall m \in R_k} l(m)}\right); \forall m \in R_k; & n = 1 \\ \min\left(1; 2\, p_{n-1}(m)\right); \forall m \in R_k; & n > 1 \end{cases} \quad (2.1)$$

where $l(m)$ is the distance between $\rho(k)$ and member $m \in R_k$. The longer is $l(m)$, the higher is the probability of m being affected by a failure. Hence, because the members affected by a failure are not known in advance, Eq. (2.1) assigns a higher probability of sending a NACK to the leaves (members) of R_k more prone to losses.

3 Performance Evaluation

Implosion is measured as the number of NACKs sent simultaneously per lost packet. Clearly, its optimal value is 1. Latency is defined as the period between packet loss detection and its successful reception at all destinations. If normalized to the timeout, its optimal value is equal to 1.

3.1 Mean Latency

Let $E[L]$ be the latency mean value. Due to the fact that the loss of a packet can occur in any link of distribution tree D, to evaluate $E[L]$, it is conditioned in the link in which the failure occur. Then, the mean latency is given by :

$$E[L] = \sum_{\forall e_{j,k} \in D} E\left[L \mid \text{failure in } e_{j,k}\right] P\left(\text{failure in } e_{j,k} \mid \text{failure}\right) \quad (3.1)$$

If it is assumed that every link has the same probability of being affected by failures, then $P\left(\text{failure in } e_{j,k} \mid \text{failure}\right) = \dfrac{1}{|L_D|}$, where $|L_D|$ is the number of links in the distribution tree D.

If we further assume that NACKs and inhibit messages are not lost, and that $E\left[L \mid \text{failure in } e_{j,k}\right]$ is calculated at the iteration at which at least one NACK is sent for the first time and the total probabilities theorem is applied, then:

$$E\left[L \mid \text{failure in } e_{j,k}\right] = \left(\sum_{n=1}^{N(A_k)} E\left[L \mid e_{j,k}, n\right] P\left(n \mid e_{j,k}\right) \right) \tag{3.2}$$

where $E\left[L \mid n, e_{j,k}\right]$ is the mean value of latency given a failure in link $e_{j,k}$ and at least one NACK sent for the first time in the n^{th} iteration of the algorithm; $P\left(n \mid e_{j,k}\right)$ is the probability that at least one NACK is sent for the first time in the n^{th} iteration of the algorithm, given a failure in link $e_{j,k} \in D$; A_k is the set of leaves of R_k affected by the failure; and $N(A_k)$ corresponds to the value of n for which at least one member of A_k sends the NACK with probability equal to 1.

To calculate $N(A_k)$, it is observed that for each leaf $m \in A_k$ there is one iteration of the algorithm in which that leaf sends a NACK with probability equal to 1. Let $N_m(A_k), m \in A_k$ be the iteration in which the leaf $m \in A_k$ reaches, for the first time, the condition $p_{N_m(A_k)}(m) \geq 1$. Eq. (2.1) implies that this condition is equivalent to $2^{N_m(A_k)-1} p_1(m) \geq 1$, which means that $N_m(A_k) \geq 1+\lg_2\left(\frac{1}{p_1(m)}\right)$. The smallest integer value that complies with the above inequality is given as $N_m(A_k) = \left\lceil 1+\lg_2\left(\frac{1}{p_1(m)}\right)\right\rceil$

Let $N(A_k)$ be the value of n for which at least one of the leaves of the tree A_k reaches the condition $p_{N_m(A_k)}(m) \geq 1$, which implies that $N(A_k) = \min_{m \in A_k} N_m(A_k)$. Then:

$$N(A_k) = \min_{m \in A_k} \left(\left\lceil 1 + \lg_2 \left(\frac{1}{p_1(m)} \right) \right\rceil \right) \tag{3.3}$$

Because failures occur with low probability, and to simplify the analysis without significantly altering the results, it is assumed that the first retransmission of the lost packet is successful. This assumption implies that:

$$E\left[L \middle| n, e_{j,k}\right] = n \cdot TO(R_k) \tag{3.4}$$

$P\left(n \mid e_{j,k}\right)$ is given by:

$$P\left(n \mid e_{j,k}\right) = \begin{cases} \left(1 - \prod_{\forall m \in A_k} \left(1 - p_1(m)\right)\right), & n = 1 \\ \left(1 - \prod_{\forall m \in A_k} \left(1 - p_n(m)\right)\right) \prod_{y=1}^{n-1} \prod_{\forall m \in A_k} \left(1 - p_y(m)\right), & n > 1 \end{cases} \tag{3.5}$$

Using Eqs. (2.1)-(3.5), E[L] can be evaluated as follows:

$$E[L] = \sum_{\forall e_{j,k} \in D} \frac{TO(R_k)}{|L(D)|} \left\{ \left(1 - \prod_{\forall m \in A_k} (1 - p_1(m))\right) + \sum_{n=2}^{\min_{m \in A_k}\left\{\left\lfloor 1 + \log_2\left(\frac{1}{p(m)}\right)\right\rfloor\right\}} n \left(1 - \prod_{\forall m \in A_k} \left(1 - p_1(m) 2^{n-1}\right)\right) \prod_{y=1}^{n-1} \prod_{\forall m \in A_k} \left(1 - p_1(h) 2^{y-1}\right) \right\} \tag{3.6}$$

3.2 Probability Mass Function of Latency

A more detailed method of characterizing latency consists of evaluating its probability mass function (pmf). Since latency is specific for each recovery tree, the pmf of the latency for a determined recovery tree is evaluated as follows:

Let $p_{T_i}(x)$ be the probability that the latency for the recovery tree T_i is equal to x. In accordance with the results obtained when $E[L]$ was evaluated, we have the following equation:

$$p_{T_i}\left(nTO(T_i)\right) = \begin{cases} \sum_{\forall e_{j,k} \in T_i} \frac{1}{c(T_i)} \left(1 - \prod_{m \in A_k} \left(1 - p_1(m)\right)\right); & n=1 \\ \sum_{\forall e_{j,k} \in T_i} \frac{1}{c(T_i)} \left(1 - \prod_{m \in A_k} \left(1 - p_1(m) 2^{n-1}\right)\right) \prod_{y=1}^{n-1} \prod_{m \in A_k} \left(1 - p_1(m) 2^{y-1}\right); & n>1 \end{cases} \tag{3.7}$$

where $c(T_i)$ is the number of links that contain the recovery tree T_i.

3.3 Mean Implosion

Let $E[I]$ be the mean value of implosion, given by:

$$E[I] = \sum_{\forall e_{j,k} \in D} E\left[I \mid \text{failure in } e_{j,k}\right] P\left(\text{failure in } e_{j,k} \mid \text{failure}\right) \tag{3.8}$$

By definition, $E\left[I \mid \text{failure in } e_{j,k}\right]$ is given by :

$$E\left[I \mid \text{failure in } e_{j,k}\right] = \sum_{n=1}^{N(A_k)} \sum_{i=1}^{|A_k|} iP\left(n, i \mid e_{j,k}\right) \tag{3.9}$$

where $P\left(n,i \mid e_{j,k}\right)$ is the probability of sending i NACKs simultaneously in the nth iteration, given a failure in link $e_{j,k}$.-

To evaluate $P\left(n,i \mid e_{j,k}\right)$, let us define $K_{u,l}$ as the set of distinct l tuples that can be formed from $u = |A_k|$ different elements, where l corresponds to the number of leaves of R_k that send a NACK. Let $\vec{k}_{u,l}$ be the u$^{\text{th}}$ l-tuple of the set $K_{u,l}$, and m the m^{th} component of $\vec{k}_{u,l}$. The probability $P\left(n,i \mid e_{j,k}\right)$ is given by:

$$P\left(n,i \mid e_{j,k}\right) = \sum_{\forall \vec{k}_{A_k,i} \in K_{A_k,i}} \left(\prod_{\forall m \in \vec{k}_{A_k,i}} p_n(m) \prod_{\forall m \notin \vec{k}_{A_k,i}} \left(1 - p_n(m)\right) \right) \prod_{y=1}^{n-1} \prod_{\forall m \in A_k} \left(1 - p_y(m)\right) \tag{3.10}$$

Then, $E[I]$ can be calculated using Eqs. (3.8)-(3.10).

4 Numerical Results

Mean latency and mean implosion were evaluated -using equations from Section 3- for network topologies of 600 nodes randomly generated using the Waxman methodology [12,13] with a 95% confidence level. Multicast groups of different sizes were randomly chosen.

Results for normalized mean latency and mean implosion as a function of the percentage of active Multicast routers in the network are shown in Fig. 1 and Fig. 2, respectively.

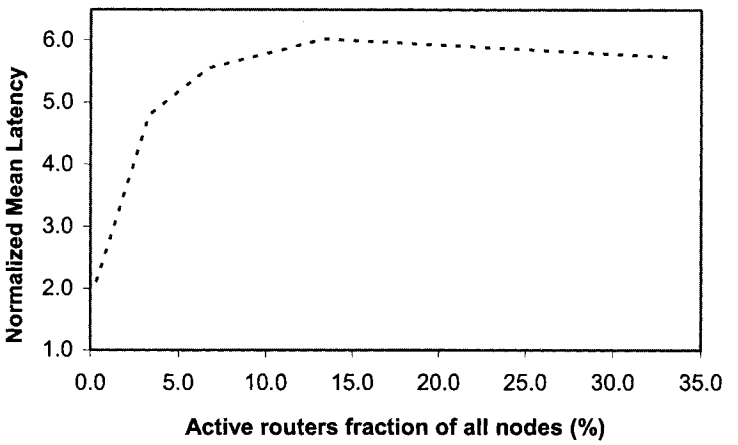

Fig. 1. Normalized mean latency versus the active routers fraction of all nodes

It can be seen that when the percentage of Multicast members in the network is less than 18%, the values for both measures increase to a maximum. However, if this percentage rises (> 18%), the distance between the location of the failure and the nearest router becomes shorter and therefore the latency to recover a lost packet declines.

As regards implosion, we observed that as the number of active routers increases, implosion remains almost unchanged and quite close to the ideal level.

These results clearly reveal the high scalability of LRARM and its ability to simultaneously achieve near-optimal implosion and very low latency.

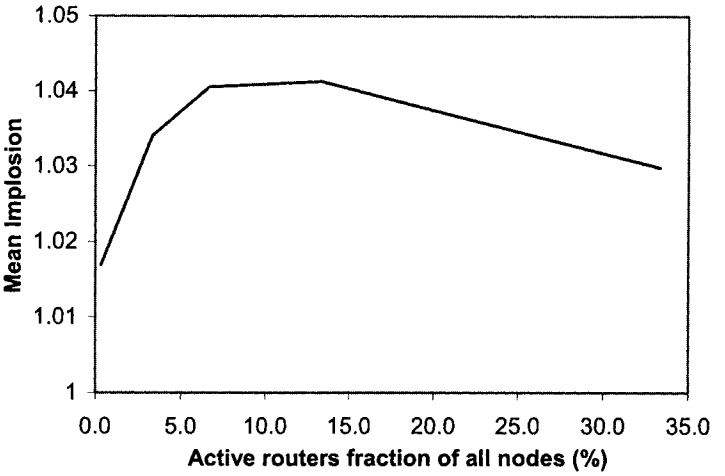

Fig. 2. Mean implosion versus the active routers fraction of all nodes

5 Conclusions

A novel error recovery algorithm for multicast transmissions, LRARM, has been proposed. LRARM simultaneously achieves near-optimal implosion and latency, regardless of Multicast group size. This makes LRARM a highly scalable algorithm with excellent performance.

Acknowledgements

Financial support from Fondecyt Project # 1000055/2000, Chile, DIPUV Project #31/2003, UTFSM Project # 230223 and Fundación Andes is gratefully acknowledged.

References

1. Adamson RB, Bormann C, Handley M, Macker J (2004) NACK-oriented reliable multicast protocol (NORM). Internet Engineering Task Force (IETF) RFC 3940
2. Diot C, DabbousW, Crowcroft J (1997) Multicast communication: a survey of protocols, functions and mechanisms. IEEE JSAC 15:277-290
3. Floyd S, Jacobson V, Liu C, McCanne S, Zhang L (1997) A reliable multicast framework for light-weight sessions and application level framing. IEEE/ACM Transactions on Networking 5:784-803
4. Gemmell J, Montgomery T, Speakman T, Bhaskar N, Crowcroft J (2003) The PGM reliable multicast protocol. IEEE Network 17:16-22.
5. Kasera S, Bhattacharyya S, Keaton M, Kiwior D, Kurose J, Towsley D, Zabele S (2000) Scalable fair reliable multicast using active services. IEEE Network Magazine 14(1): 48-57
6. Lehman L, Garland S and Tennehouse D (1998) Active reliable multicast. In: Proc. of the IEEE Infocom, San Francisco
7. Maimour M and Pham C (2002) DyRAM: a reliable multicast protocol. INRIA Report RR4635.
8. Paul S, Sabnani KK, Lin JC, Bhattacharyya S (1997) Reliable multicast transport protocol (RMTP). IEEE JSAC 15: 407-421
9. Papadopoulos C, Parulkar G, Varghese G (2004) Light-weight multicast services (LMS): a router-assisted scheme for reliable multicast. IEEE/ACM Transactions on Networking 12(3):456-468.
10. Tennehouse DL, Smith JL, Sincoskie WD, Wetherall DJ, Winden GJ (1997) A survey of active network research. IEEE Communications Magazine, pp. 80-86.
11. Towsley D, Kurose J, Pingali S (1997) A comparison of sender-initiated and receiver-initiated reliable multicast protocols. IEEE JSAC 15:398-406
12. Waxman B (1988) Routing of multicast communications. IEEE JSAC 6(9):1617-1622
13. Zegura E, Calvert K, Donahoo M (1997) A quantitative comparison of graph-based models for internet topology. IEEE/ACM Transactions on Networking 5:770-783.