ANALYSIS OF DELAY LATENCY OF THE ACTIVE RELIABLE MULTICAST PROTOCOLS

Lakhdar Derdouri*, Congduc Pham**, and Mohamed Benmohammed*

*LIRE Laboratory, University of Mentouri, Route Ain el Bey, 25000 Constantine, Algeria
derdouril@yahoo.fr, mo.esi.ben@gmail.com

**LIUPPA Laboratory, University of Pau et des Pays de l'Adour, 64013 Pau Cedex, France
congduc.pham@univ-pau.fr

ABSTRACT

This paper quantifies the reliability gain of combining classes for reliable multicasting in lossy networks in which the active network approach is the most promising. We define the delay latency of recovery as performance metric for reliability. We then study the impact of multicast group size and loss probability on the performance of compared approaches. Our simulation results show that combining classes significantly reduces the delay latency in lossy networks compared to the receiver-initiated class. Interestingly, combining classes can outperform receiver-initiated class depending on the network size and loss probability.

Keywords: *Active Networks, Reliable Multicast, Sender-Initiated, Receiver-Initiated, Delay Latency*

1. INTRODUCTION

Providing reliable and efficient multicast networking services in lossy networks is extremely challenging due to number of packets that can be corrupted or lost. To improve the reliability in lossy networks, the active networks approach has been proposed for multicast trafic.

The active networks approach provides a user driven customization of the infrastructure in which new computations are dynamically injected into the nodes [6]. The use of active networks approach in reliable multicast has proven to provide more efficient solutions to the scalability problem for a large number of receivers. In active reliable multicast protocols, the members of a multicast group are organized in a distributed control tree to overcome the well-known acknowledgment implosion problem of flat approaches, i.e., the overwhelming of the sender by a large number of positive (ACKs) or negative acknowledgments (NAKs). In addition, the concept of active network can solve the repair locality problem in an effective way by attributing the role of repair to the router close to a loss. Several active reliable multicast protocols have been proposed such as ARM (*Active Reliable Multicast*) [10], AER (*Active Error Recovery*) [8] and DyRAM (*Dynamic Replier Active reliable Multicast*) [14], AMRHy (*Active Multicast Reliable Hybrid)* [1].

AMRHy and DyRAM are two protocols that use active services within routers. Each of them adopts a different strategy to solve the scalability problems. DyRAM belongs to the receiver-initiated class where the responsibility of loss detection is attributed to receivers regardless of the link on which the losses occur. In contrast, by combining the receiver-initiated and sender-initiated classes, AMRHy distributes the responsibility of loss detection between the source and the receivers. In this hybrid approach, the source handles the losses occurring in the source link while the receivers take care for those occurring in the tail links, thus providing an efficient distribution of loss recovery burden.

This paper analyses the delay latency of the above mentioned two protocols in the presence of spatially correlated loss. Our simulation results show that the approach combining classes (AMRHy) provides good scalability and low delays compared to that based on the receiver-initiated class (DyRAM). Interestingly, combining classes can outperform the receiver-initiated class depending on the network size and loss probability.

The remainder of this paper is structured as follows: In Section 2 the existing works on analysis of reliable multicast protocols are reviewed. Section 3 presents the description of AMRHy and DyRAM protocols. Section 4 shows the network model and hypothesis. Section 5 presents the simulation results of the delay latency analysis. Conclusions and directions for future works are presented in Section 6.

2. BACKGROUND AND RELATED WORKS

The first comparative analysis of sender-initiated and receiver-initiated reliable multicast protocols was done by Pingali et al. [17]. This analysis showed thatprotocols of the receiver-initiated class are far more scalable than protocols of the sender-initiated class because the maximum throughput of the latter class is dependent on the number of receivers, while it is not the case for protocols of the receiver-initiated class. Levine et al. [11] have extended this work to ring-based and tree-based approaches and showed that the hierarchical structure organization of the receiver set in a tree-based approach guarantees scalability and improves performance. They also demonstrate that protocols based on the receiver-initiated class can not prevent deadlock when they operate with finite memory. Another comparative analysis of sender-initiated and receiver-initiated classes was presented by Maihöfer and Rothermel [12]. Their analysis showed that protocols of the receiver-initiated class achieved best scalability but those of the sender-initiated class achieved the lowest delays. Besides processing requirements, bandwidth efficiency was subject to several analytical studies. Analysis of generic reliable multicast protocols were also done by Kasera et al. [9] and their analysis showed that local recovery approaches provide significant performance increases in terms of reduced bandwidth consumption and delay. Maihöfer [13] presented an analytical bandwidth evaluation of generic reliable multicast protocols and showed that the hierarchical approaches provide higher throughput as well as lower bandwidth consumption. A throughput analysis of reliable multicast protocols in an active networking environment was done by Maimour and Pham [15] and this analysis showed that the achievable throughput increases as the number of active routers increases.

On the other hand,most active reliable multicast protocols adopt a local recovery approach which is based on the receiver-initiated class, e.g. ARM, AER, DyRAM.This class has several advantages:

1. The source does not know the receivers set.
2. The source does not have to process ACKs from each receiver.
3. The receivers pace the source.

However it also suffers from some restrictions:

1. A high recovery latency that is not acceptable for some real time applications which require not only reliability but also the lowest delay latencies.
2. An inefficient distribution of the loss recovery burden between the source and the receivers: losses occurring on the links close to the source will be detected only at the leaves of the multicast tree by the receivers.
3. An inefficient management of the active routers cache: there is no means for the active routers to know when they can safely release data from their cache.
4. The risk that a data packet never reaches its destination when the source has a limited number of buffers in emission.
5. The election time of the replier can become too considerable when the NAKs are lost: the active router must make several attempts to elect the adequate replier.

Combining classes can cure these disadvantages and would therefore allow an efficient distribution of the loss recovery burden between the source and the receivers. Consequently we have proposed AMRHy [1] which combines the 'sender-initiated' and the 'receiver-initiated' classes. In [2] we presented an analytical study comparing the combination of classes with receiver-initiated class, and we showed that combining classes provides higher throughput and lower usage of bandwidth. We showed that combining classes perfectly adapt to unreliable environments and offer better scalability in a large group of receivers. In this paper we extend the previous work by comparing AMRHy with DyRAM in terms of the delay latency of data delivery in active networking environments.

3. PROTOCOLS DESCRIPTION

The first considered protocol is based on the receiver-initiated class (DyRAM). Receiver-initiated protocols return only NAKs from receivers to sender instead of ACKs. A receiver experiencing a packet loss returns a NAK to the sender. DyRAM uses global suppression of NAKs. The active routers have in charge the aggregation of NAKs in order to forward only one NAK to the sender.

The second considered protocol is a combination of sender-initiated and receiver-initiated classes (AMRHy). The combination of classes returns both ACK and NAK. The ACK has global meaning; it is used between the sender and a receiver to report the successful reception of data. It permits the sender to release the corresponding buffer space and to adjust the emission window. It also permits an active router to:

1. Inform the remainder of its local group having received a data packet to locally suppress their ACKs.
2. Inform the remainder of its local group having lost a data packet of its availability in its cache and also to communicate the address of the replier for future repair without using active services.
3. Release a corresponding buffer space.

The NAK is used locally between the receivers and their active routers for requesting a lost data packet. AMRHy uses both global and local suppression of ACKs. The active routers have in charge the aggregation and suppression of ACKs in order to forward only one ACK to the sender.

Yeung et al. [19] have defined the taxonomy of reliable multicast protocols in which protocols are grouped according to the following two criteria: sender-initiated or receiver-initiated, hierarchical-based or timer-based. Figure 1 shows the position of AMRHy and DyRAM in this protocols classification.

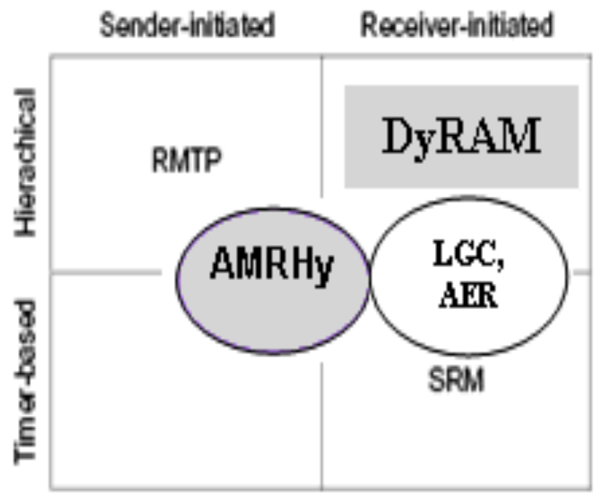

Figure 1: AMRHy and DyRAM in the reliable multicast protocols classification.

3.1. DESCRIPTION OF DyRAM

DyRAM protocol exhibits the following behavior [14]:

1. The sender multicasts data packets to a multicast address that is subscribed to by all receivers.
2. Upon reception of a NAK, the sender multicasts data packets to a multicast address that is subscribed to by all receivers.
3. Upon reception of a data packet, an active router stores it in its cache, when possible, and forwards it downstream in the multicast tree.
4. Upon reception of a repair packet, an active router subcasts it downstream to receivers having requested it.
5. Upon detection of a packet loss, an active router immediately sends a NAK towards its ascendant in the multicast tree and sets a timer.
6. On timeout, an active router sends a NAK towards its ascendant in the multicast tree and sets a timer.
7. Upon reception of a valid NAK, an active router sets a DTD timer which permits the replier election.
8. On DTD timeout, an active router sends a NAK towards an elected replier if it exists; otherwise it sends a NAK towards the sender.
9. Upon detection of a packet loss, a receiver immediately returns a NAK towards the sender and sets a timer.
10. Upon reception of a NAK, receiver sends the requested packet if it is available, otherwise it sends a NAK to its active router.

3.2. DESCRIPTION OF AMRHy

AMRHy protocol exhibits the following behavior [1]:

1. The sender multicasts data packets to a multicast address that is subscribed to by all receivers, and sets a timer.
2. On timeout, the sender multicasts data packets to a multicast address that is subscribed to by all receivers, and sets timer.
3. Upon reception of an ACK, the sender releases a corresponding buffer space and adjusts its emission window.
4. Upon reception of the first ACK from a descendant, an active router dispatches the ACK to the other receivers in its local group and sets a WP timer before forwarding it to its ascendant in the multicast tree. During this period, it ignores all the duplicate ACKs from the descendants.
5. Upon reception of an ACK from an ascendant during the waiting period, an active router verifies if the corresponding data packet was received. If so, it behaves as it has sent an ACK; otherwise it sends a NAK to its ascendant in the multicast tree and sets a timer.
6. On timeout, an active router sends a NAK to its ascendant in the multicast tree and sets a timer.
7. When the WP timer expires at an active router, if it has not received an ACK from the ascendant, the active router would send an ACK towards its ascendant in the multicast tree; subcast the data packet towards receivers having requested it and release a corresponding buffer space.
8. Upon reception of a NAK from an ascendant or descendant, an active router sends the requested data packet if it is available in its cache, otherwise it forwards a NAK to the replier (a receiver which has sent the first ACK).
9. Upon reception of a repair packet, an active router forwards it to the nodes having requested it (ascendant or descendant).
10. Upon reception of a data packet, a receiver waits for a random period before sending an ACK to its active router. If during this waiting period, a receiver receives an ACK then it behaves as if it has send an ACK.
11. On timeout, a receiver sends an ACK towards its active router.
12. Upon reception of an ACK, a receiver verifies the corresponding data. If it has already been received then the receiver behaves as if it has sent an ACK to its active router, otherwise it sends a NAK to its active router and sets a timer.

4. NETWORK MODEL AND HYPOTHESIS

A commonly used model for evaluating multicast protocols is to have a multicast tree rooted at the source with receivers as leaves (see Figure. 2). Intermediate nodes are routers. In the context of active networking, we consider that the active routers are placed at strategic points within the network where the losses often occur. These points are usually located at the edge of the backbone for two essential reasons:

1. The backbone is supposed to be reliable: [18] showed that the links where most of losses occurred are those located at network's edge.
2. The backbone is a very high-speed network and adding complex processing functions inside the backbone will certainly degrade its performance.

Consequently these active routers are able to perform customized processing such as the aggregation/suppression of the acknowledgement packets and the cache of data packets for the local recovery of the losses.

Our study is based on the following assumptions:

1. For the loss model, we consider that the core network is reliable, as mentioned previously. For the links (the source link and tail link), the loss is noted p_l. Therefore, the end to end probability of a packet loss perceived by receiver is $p = 1-(1- p_l)^2$. The losses are assumed to be temporally independent and those at the tail links are assumed to be mutually independent.
2. The links between the active routers are identical (the same theoretical throughput).
3. The links between the active routers and the receivers are identical (the same theoretical throughput).

Once the topology was defined, the next step is to define the behaviour of the various elements of the network (source, active routers and receivers). The behaviour of each element is the result of the interaction between the various protocols of the protocols stack. Thus, to define the behaviour of an element of the network, it is necessary to determine the protocols that are used at each layer. We have chosen to use the PIM-SM protocol as a multicast routing protocol at the network layer [3]. Since selected topology is not dynamic and multicast groups do not undergo any change, the use of PIM-SM protocol minimizes the transit of routing packets in the network which in turn minimizes the influence of these packets in the study. Once the routing protocol is selected, it is important to define the various parameters to be set up in the implementation of the network elements. These parameters allow us to estimate the performances of each protocol.

In our study, only the delay latency metric is used to determine the performances of each reliable protocol multicast. This metric expresses the average time required to transmit in a reliable way a data packet from the source to a receiver. The analysis of the delay latency enables us to determine which of the two protocols is best adapted to the applications transmitting data with real time constraints.

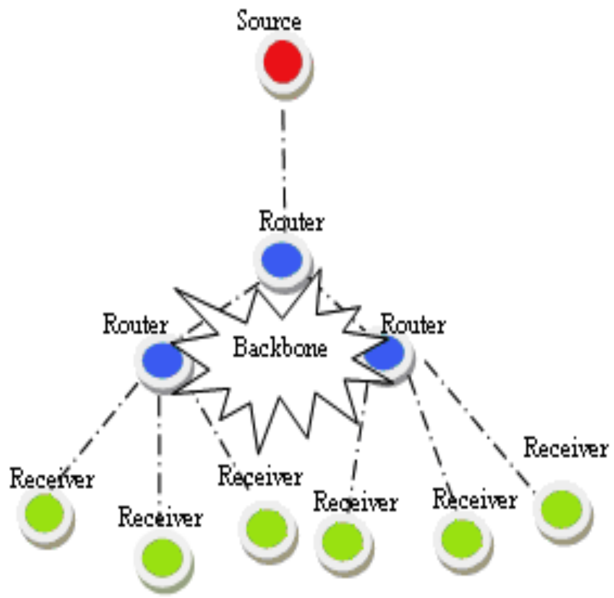

Figure 2: Network model.

5. SIMULATION RESULTS

In this section, we expose results of simulations obtained after having implemented AMRHy and DyRAM protocols in the NS2 environment [7]. We evaluate the delay latency by comparing the average time each packet takes to reach the destination. The results are presented according to the impact of the loss probability and the multicast group size. For our study we fixed the following values:

1. The transfer time of a packet from the source to the close active router is 0.02 ms.
2. The transfer time of a packet from an active router to an other active router is 0.05 ms.
3. The transfer time of a packet from an active router to a receiver is 0.05 ms.

5.1 IMPACT OF LOSS PROBABILITY

We study for each protocol the average delay for a data packet to be received by a randomly chosen receiver according to the loss probability. The delay includes the time required to detect the loss and the time required to perform the recovery. The group size is fixed to 100 receivers.

Figure 3 shows that for low loss probabilities DyRAM allows a faster delivery of the data packets than AMRHy. However, we can see that the benefit of AMRHy over DyRAM increases rapidly as the loss probability increases.

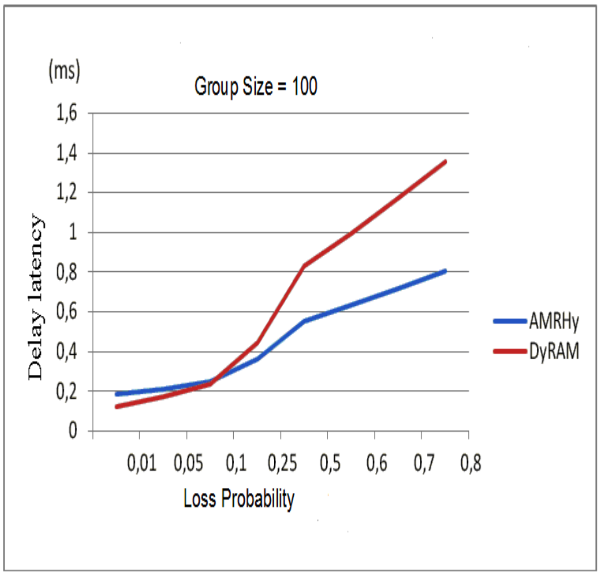

Figure 3: Impact of loss probability on delay latency.

5.2 IMPACT OF GROUP SIZE

After having studied the impact of the loss probability on the performance of both protocols, we present a comparison of AMRHy and DyRAM according to the multicast group size. We set the loss probability to p=0.1.

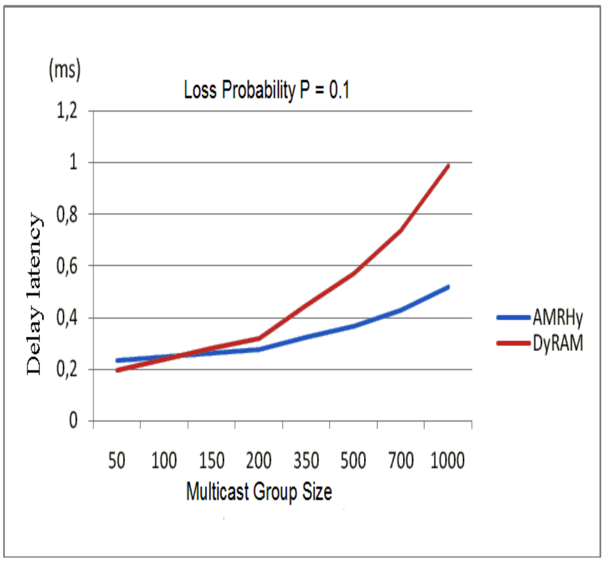

Figure 4: Impact of group size on delay latency.

Similarly, AMRHy presents a lower delay with respect to the deployment of multicast groups compared to DyRAM (see Figure 4). The larger the multicast group size, the larger the delay gap between AMRHy and DyRAM. The performance degradation of DyRAM is due to the inefficient distribution of the loss recovery burden in the receiver-initiated class which attributes the losses detection to the receivers regardless of the link on which the losses occur. If a loss occurs on the source link, all the receivers are requested to seek the lost packet from the source causing an important delay. On the other hand, when combining classes this kind of loss is detected by the source. This result confirms that combining classes is more scalable in unreliable environments than the receiver-initiated class alone.

6. CONCLUSION

In this paper, we studied the benefits of combining classes for active reliable multicasting. We used the delay latency of recovery as the performance metric of interest. We studied the impact of different parameters such as the loss probability and the multicast group size on the performance. We found that combining classes (AMRHy) significantly reduces the delay latency of recovery in lossy networks compared to the DyRAM receiver-initiated protocol. The simulation results demonstrated that combining classes can achieve high reliability while saving network resources. The performance gains increase as the size of the network and the loss probability increase which make the combination of classes more scalable with respect to these parameters. In the future, we would like to extend our study to more complex multicast topologies with a significant amount of path diversity.

REFERENCES

[1] L. Derdouri, C. Pham, M. Benmohammed, AMRHy: Hybrid Protocol for Reliable Multicast Transport in Active Networking Environment, *International Journal on Computer Science and Network Security* (*IJCSNS*), vol. 8(10), pp.68-78, October, 2008.

[2] L. Derdouri, C. Pham, M. Benmohammed, A Comparative Analysis of Reliable Multicast Protocols in Active Networking Environments, *International Review on Computers and Software* (*IRECOS*), vol. 3(3), pp.265-275, Praize Worthy Prize, May 2008..

[3] D. Estrin et al., Protocol Independent Multicast Sparse Mode (PIM-SM), Protocol Specification, RFC 2362, June 1998.

[4] S. Floyd, V. Jacobson, S. Mc Canne, C-G. Liu, L. Zhang, A Reliable Multicast Framework for Light-Weight Sessions and Applications Level Framing, *IEEE/ACM Transactions on Networking*, vol. 5(6), pp. 784-803, 1997.

[5] M. Hofmann, Enabling Group Communication in Global Networks, *Proceeding Global* Networking'97, pp. 321, Calgary, Alberta, Canada, June 1997.

[6] S. A. Hussain, *Active and Programmable Networks for Adaptive Architectures and Services*, Auerbarch Publication, Taylor & Francis group, an informa business, 2007.

[7] http://www.isi.edu/nsnam/ns/index.html.

[8] S. K. Kasera et al., Scalable Fair Reliable Multicast Using Active Services, *in Network, IEEE,* vol. 14(1) pp. 48-57, August 2002.

[9] S. Kasera, J. Kurose, D. Towsley, A Comparison of Server-based and Receiver-based Local Recovery Approaches for Scalable Reliable Multicast, *in proceeding of the 17th IEEE INFCOM'98*, pp. 988-995, April 1998.

[10] L. H. Lehman, S. J. Garland, D. L. Tennenhouse, Active Reliable Multicast, *in proceeding of INFCOM'98*, pp. 512, San Francisco, USA, 1998.

[11] B. Levine, J. J. Garcia-Luna-Aceves, A Comparison of Reliable Multicast Protocols, *Multimedia systems*, vol. 6(6), pp. 334-348, 1998.

[12] C. Maïhofer and K. Rothemel, A Delay Analysis of Tree-based Reliable Multicast Protocols, *in proceeding of the Tenth International Conference on Computer Communication and Networks,* pp. 274-281, Scottsdale, USA, October, 2001.

[13] C. Maiföfer, A Bandwidth Analysis of Reliable Multicast Transport Protocols, *proceeding of NGC 2000 on Networked Group Communication*, pp. 15-26, Stanford, November, 2000.

[14] M. Maimour and C. Pham, DyRAM: an Active Reliable Multicast Framework for Data Distribution, *Journal of Cluster Computing*, vol. 7(2), pp. 407-421, 1997.

[15] M. Maimour and C. Pham, A Throughput Analysis of Reliable Multicast Protocols in Active Networking Environment, *proceeding of the sixth IEEE Symposium on Computers and Communications*, pp. 151-158, July 2001.

[16] S. Paul, K. Sabnani, J. C. Lin, S. Bhattacharyya, Reliable Multicast Transport Protocol (RMTP), *IEEE Journal on Selected Areas in Communications*, Vol. 15(3), pp. 407-421, 1997.

[17] S. Pingali, D. Towsley and J. K. Kurose, A Comparison of Sender-initiated and Receiver-initiated Reliable Multicast Protocols, *proceeding ACM SIGMETRICS94*, pp. 221-230, October 1994.

[18] M. Yajnik, J. Kurose, and D. Towsley, Packet Loss Correlation in the Mbone Multicast Network, *proceeding of Global Telecommunication Conference*, pp. 94-99, November 1994.

[19] K.. L. Yeung, H. T. Wong, Caching Policy Design and Cache Allocation in Active Reliable Multicast, *Computer Networks*, vol. 43(2), pp. 177-193, 2003.