Analysis of Hybrid Router-Assisted Reliable Multicast Protocols in Lossy Networks

Lakhdar Derdouri[1], Congduc Pham[2], and Mohamed Benmohammed[3]
[1]Math and Computer Science Department, Larbi Ben M'hidi University, Algeria
[2]Computer Science Department, Pau University, France
[3]Faculty of New Technology of Information and Communication, Abdelhamid Mehri Constantine 2 University, Algeria

Abstract: *Router-assisted concepts have been proposed in many research areas including reliable multicast protocols. These concepts can limit the implosion and repair locality problems in an effective way by attributing the role of repair locality to the specific router close to the point of packet loss. Several router-assisted reliable multicast protocols have been proposed in the literature. However, the extent of the reliability benefit of combining sender-initiated and receiver-initiated protocol classes is not known. This paper quantifies the reliability gain of combining classes for reliable multicasting in lossy networks. We define the delivery delay, the bandwidth consumption, and the buffer requirements as the performance metrics for reliability. We then use simulations to study the impact of multicast group size and loss rate on the performance of combining protocol classes. Our numerical results show that combining classes significantly improves the delivery delay, reduces the consumption of the network bandwidth and minimizes the buffer size at the routers compared to receiver-initiated class alone. The performance gains increase as the size of the network and the loss rate increase, making the combination of classes approach more scalable with respect to these parameters.*

Keywords: *Router-assisted, reliable multicast, sender-initiated, receiver-initiated, delay, bandwidth, buffer requirements.*

Received September 10, 2013, accepted May 20, 2014

1. Introduction

Multicast is an efficient way to disseminate data to a group of receivers that are interested in the same content. It provides an efficient means of supporting collaborative applications such as videoconferencing, distributed gaming, distance learning, IP-TV and Video-On-Demand (VOD), etc. Multicast naturally fits such applications by constructing a routing multicast tree which allows the source to simultaneously reach all the receivers. Especially, it helps to reducing the bandwidth consumption in environments, in which this one is considered as a scarce resource like wireless environments. Multicast can represent a huge enhancement of the network capacity by taking advantage of links that can be shared by multiple users to receive the same data, which is transmitted only once. Besides the effectiveness of the routing layer, most collaborative applications have a strict requirement of 100% Packet Delivery Ratio (PDR), since every byte of the downloaded file has to be received by all the receivers. Providing reliable and efficient multicast networking services in lossy networks is extremely challenging due to the number of packets that can be corrupted or lost.

Govindaswamy and Muthusamy [10] have proposed a reliable broadcasting algorithm, for mobile ad-hoc networks, using the efficient forwarding router selection mechanism. Each router stores the packet, calculates it's forwarding routers, and rebroadcasts the packet as a new sender, after receiving a new broadcast packet. The sender eavesdrop the retransmissions of the forwarding routers as the acknowledgement of receiving the packet. This algorithm reduces the average retransmission redundancy, avoids both the broadcast storm problem and the ACK implosion problem, and locally recovers the transmission errors.

The active networks model provides a user driven customization of the infrastructure in which new computations are dynamically injected into the routers. It has the ability to provide a very general and flexible framework for customizing network functionalities in order to gracefully handle heterogeneity and traffic dynamics [9]. The idea of programmable networks has recently regained considerable momentum due to the emergence of the Software-Defined Networking (SDN) paradigm [1]. This novel architecture decouples the network control and forwarding functions. It enables the network control to become directly programmable and the underlying infrastructure to be abstracted for varied applications. While not being fully implemented, programmable network researches have paved the way to so-called router-assisted approaches and router-assisted multicast has proven to provide more efficient solutions to the scalability problems [2, 3, 8, 15].

In most router-assisted 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 (NAKs) acknowledgments. In addition, the router-assisted approach can solve the repair locality problem in an effective way by attributing the role of repair to the router close to the point of packet loss. Several router-assisted reliable multicast protocols have been proposed such as Active Reliable Multicast (ARM) [13], Active Error Recovery (AER) [12] and Dynamic Replier Active reliable Multicast (DyRAM) [19], Active Multicast Reliabl**e** Hybrid (AMRHy) [5].

AMRHy and DyRAM are two protocols that use propose additional 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. AMRHy adopts a hybrid approach based on the combination of sender-initiated and receiver-initiated classes. In this hybrid approach, the source handles packet losses occurring in the source link while the receivers take care for those occurring in the tail links, which allows an efficient distribution of loss recovery burden.

In this paper, we evaluate the two mentioned protocols in the presence of spatially correlated loss throughout 3 metrics: The bandwidth consumption, the delivery delay and the buffer requirements. Simulations show that the approach combining classes AMRHy outperforms those based on the receiver-initiated class DyRAM. Interestingly, this result shows the need of combining classes in designing protocols that provide high scalability in lossy networks such as wireless environments.

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 delivery delay, the bandwidth consumption and the buffer requirements analysis. Conclusions and directions for future works are presented in section 6.

2. Related Works

The first comparative analysis of sender-initiated and receiver-initiated reliable multicast protocols was done by Pingali *et al.* [21]. This analysis showed that protocols 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 and Garcia-Luna-Aceves [14] 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 cannot prevent deadlock when they operate with finite memory. Another comparative analysis of sender-initiated and receiver-initiated classes was presented by Maihöfer and Rothemel [16]. 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.* [11] and their analysis showed that local recovery approaches provide significant performance increases in terms of reduced bandwidth consumption and delay. Maihöfe [17] 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 [18] and their analysis showed that the achievable throughput increases as the number of router-assisted increases.

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

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

However, it also suffers from some restrictions:

- High Recovery Latency: More real time collaborative applications require not only the reliability but also the lowest delivery latencies.
- Inefficient Distribution of the Loss Recovery Burden: Losses occurring on the links close to the source will be detected at the leaves of the multicast tree by the receivers.
- Inefficient Management of the Routers Cache: The routers do not know when they can safely release data packets from their cache.
- The Risk that a Data Packet Never Reaches: Its destination when the source has limited buffers in emission.

- Important Election Time of the Replier: When NAKs are lost the router makes several attempts to elect the adequate replier.

Combining classes can remedy these restrictions and would therefore allow an efficient distribution of the loss recovery burden between the source and the receivers. Therefore, we propose AMRHy [5] that combines the sender-initiated and the receiver-initiated classes. In [4] we presented an analytical study comparing the combination of classes with the receiver-initiated class, and we showed that combining classes provides higher throughput and lower usage of bandwidth. We also showed that combining classes perfectly adapts to unreliable environments and offers better scalability. In this paper, using simulation, we extend the previous work by comparing AMRHy with DyRAM in terms of the delivery delay [6], consumed bandwidth which validates our previous analytical result presented in [4], and the buffer requirements at the routers having the responsibility of supplying repairs to their downstream receivers.

3. Description of Protocols

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 routers have in charge the aggregation of NAKs in order to forward only one NAK to the sender, the best effort cache of data packet for future local recovery, and dynamic election of the replier among receivers in order to establish a balanced load of loss recovery within a local group.

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 a router to:

- Inform the remainder of its local group having received a data packet to locally suppress their ACKs.
- 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 additional services.
- Release a corresponding buffer space.

The NAK is used locally between the receivers and their close routers for requesting a lost data packet. AMRHy uses both global and local suppression of ACKs. The routers have in charge the aggregation and local suppression of ACKs in order to forward only one ACK to the sender, the cache of data packet for a well defined period in order to ensure local recovery, and dynamic election of the replier among receivers in order to establish a balanced load of loss recovery within a local group.

Yeung and Wong [23] 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.

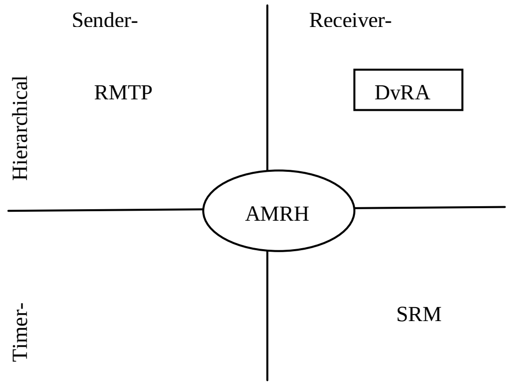

Figure 1. AMRHy and DyRAM in reliable multicast protocols classification.

3.1. Description of DyRAM

DyRAM exhibits the following behavior [19]:

- The sender multicasts data packets to a multicast address that is subscribed to by all receivers.
- Upon reception of a NAK, the sender multicasts data packets to a multicast address that is subscribed to by all receivers.
- Upon reception of a data packet, a router stores it in its cache, when possible, and forwards it downstream in the multicast tree.
- Upon reception of a repair packet, a router subcasts it downstream to receivers having requested it.
- Upon detection of a packet loss, a router immediately sends a NAK towards its ascendant in the multicast tree and sets a timer.
- On timeout, a router sends a NAK towards its ascendant in the multicast tree and sets a timer.
- Upon reception of a valid NAK, a router sets a Delay To Decide (DTD) timer which permits the replier election.
- On DTD timeout, a router sends a NAK towards an elected replier if it exists; otherwise it sends a NAK towards the sender.
- Upon detection of a packet loss, a receiver immediately returns a NAK towards the sender and sets a timer.
- Upon reception of a NAK, receiver sends the requested packet if it is available, otherwise it sends a NAK to its router.

3.2. Description of AMRHy

AMRHy exhibits the following behavior [5]:

- The sender multicasts data packets to a multicast address that is subscribed to by all receivers, and sets a timer.
- On timeout, the sender multicasts data packets to a multicast address that is subscribed to by all receivers, and sets timer.
- Upon reception of an ACK, the sender releases a corresponding buffer space and adjusts its emission window.
- Upon reception of the first ACK from a descendant, a router dispatches the ACK to the other receivers in its local group and sets a Waiting Period (WP) timer before forwarding it to its ascendant in the multicast tree. During this period, it ignores all the duplicate ACKs from the descendants.
- Upon reception of an ACK from an ascendant during the waiting period, a 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.
- On timeout, a router sends a NAK to its ascendant in the multicast tree and sets a timer.
- When the WP timer expires at a router, if it has not received an ACK from the ascendant, the router would send an ACK towards its ascendant in the multicast tree; subcasts the data packet towards receivers having requested it and release a corresponding buffer space.
- Upon reception of a NAK from an ascendant or descendant, a 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).
- Upon reception of a repair packet, a router forwards it to the nodes having requested it (ascendant or descendant).
- Upon reception of a data packet, a receiver waits during a random period before sending an ACK to its router. If during this waiting period, a receiver receives an ACK then it behaves as if it has sent an ACK.
- On timeout, a receiver sends an ACK towards its router.
- 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 router, otherwise it sends a NAK to its 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 as shown in Figure 2. Intermediate nodes are routers. In the context of router-assisted protocols, the routers are able to perform customized processing such as the aggregation/suppression of the acknowledgements and the cache of data packets for the local recovery of losses. They are placed at strategic locations inside the network where the losses often occur.

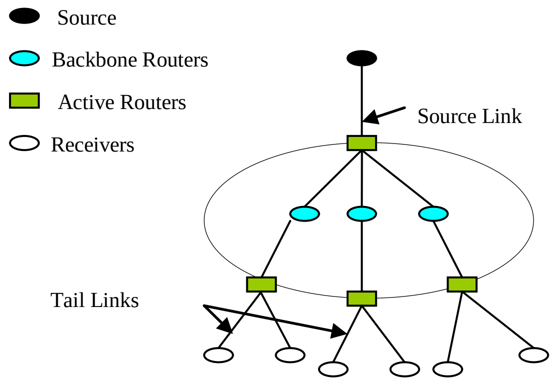

Figure 2. Network model.

These locations are usually located at the edge of the backbone for two essential reasons:

- The Backbone is Supposed to be Reliable: Yajnik *et al.* [22] have observed that most of the losses take place in the links located at network's edge.
- The backbone is a very high-speed network and adding complex processing functions inside it will certainly degrade its performance.

Our study is based on the following assumptions:

- For the loss model, we consider that the core of the network is reliable, as mentioned previously. For the links (the source links and tail links), 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 at source links are assumed to be temporally independent and those at the tail links are assumed to be mutually independent.
- The links between the routers are identical (the same theoretical throughput).
- The links between the routers and the receivers are identical (the same theoretical throughput).

Once the topology was established, the following step is to define the behaviour of each node of the multicast tree (source, routers and receivers). This behaviour is the result of the interaction between different protocols of all layers that constitute this node. Thus, it is important to determine the protocol that is used at each layer. We chose to use the PIM-SM [24] as a multicast routing protocol at the network layer. Since, the selected topology is not dynamic and multicast groups do not undergo any change, the use of PIM-SM minimizes the traffic of routing packets in the multicast tree which in turn minimizes the influence of these packets on our study. Therefore, it is important to define the various parameters to be set up in the implementation of the nodes of the multicast tree. These parameters allow us to estimate the performance of each protocol. In our study, we evaluate the reliability gain of combining classes by comparing AMRHy and DyRAM through the following metrics.

- Bandwidth: The mean number of packets transmitted in the network.
- Delay: The average time required to transmit in a reliable way a data packet from the source to a receiver.
- Buffer Requirements: The amount of memory a router uses to buffer packets from the source in order to be able to later retransmit them to repair downstream losses.

We use the NS2 simulator [20] to simulate a hierarchical multicast tree (illustrated in Figure 2.) varying the multicast group size (the number of receivers per group from 1000 to 100000) and the packet loss rate (the loss probability varies from 0.01 to 0.8). The multicast source generates traffic at Constant Bit Rate (CBR) properly set for each scenario. The packet size for all traffic is set to 512bytes for data packets and 32bytes for ACK/NAK packets. The simulation time is 60seconds. We vary the transmission rate from 50packets/s to 1000 packets/s, and measure the delay, the bandwidth and the buffer size.

5. Analysis of Simulation Results

In this section, we expose results of simulation obtained after having implemented AMRHy and DyRAM in the NS2 environment.

5.1. Bandwidth Analysis

We analyze the network bandwidth requirements of AMRHy and DyRAM. The analysis of the traffic generated by each protocol allows us to determine its requirements in bandwidth. For that, we need to find the mean number of packets (data, repairs, and acknowledgement) flowing through the multicast tree during a multicast session by respectively using AMRHy and DyRAM. Our study shows that this number is influenced by two essential parameters: The packet loss rate and the multicast group size.

1. Impact of Packet Loss Rate: We study the need for each protocol in terms of consumed bandwidth according to the loss rate.

The result obtained in Figure 3; shows that DyRAM presents a better usage of bandwidth than AMRHy for small loss rates. However, for high loss rates, the performance of DyRAM is subject to a considerable degradation while AMRHy's performance remains constant. This result can be interpreted by the fact that when the loss rate increases, the risk for losses to occur on the source link is very high, decreasing the number of the ACKs announcing the successful reception of the data packet. On the other hand, in combining classes, losses that occur in the tail link are indicated when receiver receives an ACK numbered by not received packet. For high loss rates, as less ACK are received then less NAK are generated, thereby reducing the bandwidth usage. We can conclude that AMRHy adapts perfectly with the unreliable environments whereas DyRAM presents satisfactory performance only for reliable networks.

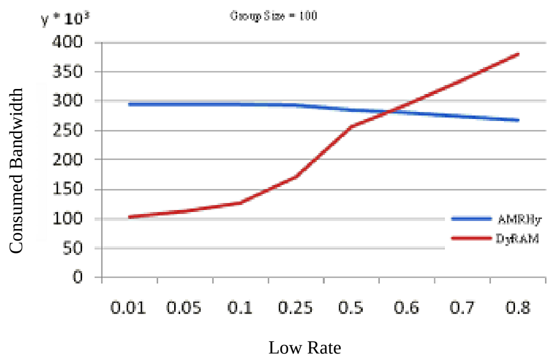

Figure 3. Impact of packet loss rate on the consumed bandwidth.

2. Impact of Multicast Group Size: We study the bandwidth usage of each protocol according to the multicast group size. We make the comparison by assigning to the loss rate 3 different values p=0.1, p=0.5, p=0.6.

The results illustrated in Figures 4, 5 and 6 show that for reduced group size and a low loss rate AMRHy generates more traffic than DyRAM. This result can be interpreted by the fact that AMRHy uses much more bandwidth than DyRAM because it is based on the explicit transmission of the ACK for each transmitted packet, whereas DyRAM is based on the ACK implicitly transmitted within NAK. It is important to notice that our study quantifies the consumed bandwidth in terms of number of packets forwarded through the multicast tree and that no consideration was related to the packets size; AMRHy's control packets are very small compared to those of DyRAM that contain several fields and can be rather voluminous [7, 19]. However, DyRAM does not provide the same performance level for larger multicast group size with a loss rate higher than 50%. This is mainly due to the inefficient distribution of loss recovery burden between the source and the receivers, where losses occurring on the source links will be detected by the receivers, thus provoking an important feedback traffic which will be repeated until the lost data packet will correctly be received by all the receivers.

We can conclude that as opposed to DyRAM, AMRHy significantly limits the network bandwidth in unreliable environments with large multicast group size. The combination of classes establishes a more efficient distribution of loss recovery burden succeeds in reducing the feedback flow generated when the loss occurs on the source link.

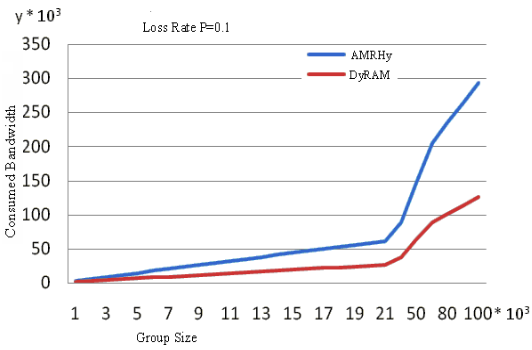

Figure 4. Impact of multicast group size on the consumed bandwidth (p= 0.1).

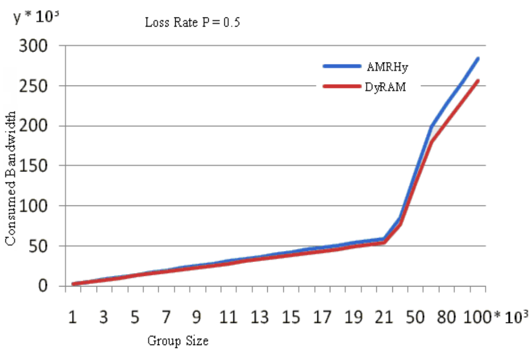

Figure 5. Impact of multicast group size on the consumed bandwidth (p= 0.5).

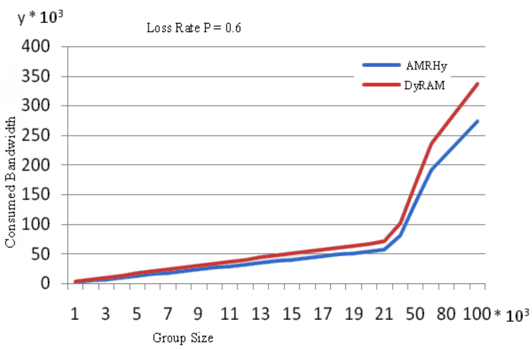

Figure 6. Impact of multicast group size on the consumed bandwidth (p= 0.6).

5.2. Delay Analysis

We evaluate the delivery delay of AMRHy and DyRAM by comparing the average time allowing a packet to reach the destination. This delay enables us to determine which of the two protocols is best adapted to the applications transmitting the data with real time constraints. For our study we fixed the following values:

- The transfer time of a packet from the source to the closest router is 0.02 ms.
- The transfer time of a packet from a router to another router is 0.05 ms.
- The transfer time of a packet from a router to a receiver is 0.05 ms.

3. Impact of Packet Loss Rate: We study, for each protocol, the average delay for a data packet to be received by a randomly chosen receiver according to the loss rate. 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 7 shows that for low loss rates, 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 rate increases.

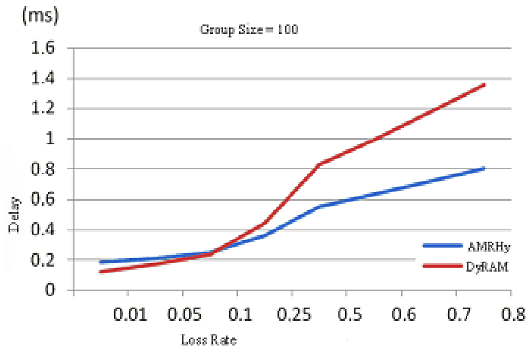

Figure 7. Impact of packet loss rate on the delay.

4. Impact of Multicast Group Size: After having studied the impact of the loss rate 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.

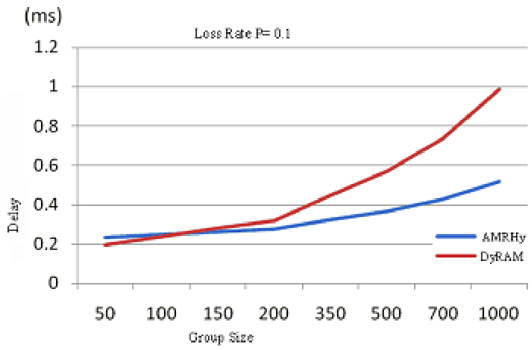

Figure 8. Impact of multicast group size on the delay.

Similarly, AMRHy presents a lower delay with respect to the deployment of multicast groups compared to DyRAM as shown in Figure 8. The larger the multicast group size, the larger the delay gap between AMRHy and DyRAM. The performance degradation of DyRAM is due mainly to the inefficient distribution of the loss recovery burden in the receiver-initiated class. This class 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. In AMRHy this category of losses is detected by the source. This result confirms that combining classes is more scalable in unreliable environments than the receiver-initiated class alone.

5.3. Buffer Requirements Analysis

One main task of a router in a router-assisted approach is to buffer the source's data packets in order to later repair downstream losses. Theoretically, in a receiver-initiated class such as DyRAM, a router needs to buffer each packet for an undefined amount of time in order to retransmit a packet whenever a NAK for that packet is received. Such local recovery behaviour requires an unlimited buffer space at the router. Practically, a multicast session will be allocated a certain amount of buffers; when all its buffers are full, a buffer replacement policy will be used to replace old packets in the buffer with new packets arriving from the source. It is therefore possible that some of the packets will be removed from the buffer of the router before they are successfully received downstream. These packets must then be recovered upstream from the router. The advantage of combining classes is that the source and the routers know when they can safely release data from the buffers.

In the following simulations we examine the buffer requirements of a router for each protocol according to the number of packets transmitted through the multicast tree. We study the impact of the multicast group size and the loss rate on the performances of AMHRy and DyRAM.

5. Impact of Packet Loss Rate: We study the requirements in terms of the storage capacity for each protocol according to the loss rate.

Figure 9 shows that AMRHy still presents a constant and positive behaviour with respect to the considerable degradation in the reliability of the transmission network, whereas DyRAM has a strong sensibility to this factor. This behaviour is explained by the fact that DyRAM keeps in its cache the list of the lost packets by each one of its descendant. The more the number of losses the larger the space dedicated to packet storage. We note that in DyRAM the packet's suppression time from the cache of the router remains unknown. AMRHy is not sensitive to the loss rate because it stores the lost packets in the cache of the router only for a fixed period of time proportional to the RTT to the farthest descendant from the router.

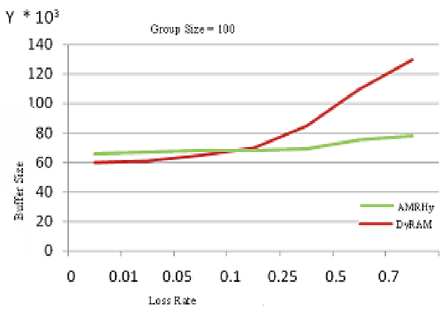

Figure 9. Impact of packet loss rate on the buffer requirements.

6. Impact of Multicast Group Size: Figure 10 shows that AMRHy always presents a constant storage space when the multicast group size increases while this is not the case for DyRAM. Once again, we fixed the loss rate to p=0.1.

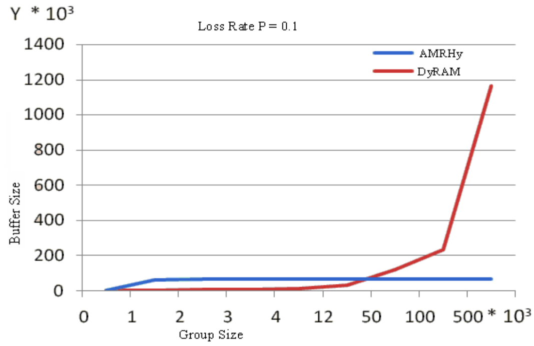

Figure 10. Impact of multicast group size on the buffer requirements.

This result is due to the fact that besides the data packet's suppression time that remains unknown, DyRAM also keeps in the cache of the router the state of its descendants in 2 different logic structures. Whereas, AMRHy keeps the data packet in the cache of the router only for a limited period during which it knows all the receivers having really lost the data packet. After this time period, the packet will be removed from the cache. We can conclude that AMRHy shows a more suitable behaviour for unreliable networks with large multicast group size by greatly reducing buffer size in routers.

6. Conclusions

In this paper, we studied the reliability benefits of combining classes for reliable multicasting through lossy networks such as wireless environments. Combining classes provides the ability to quickly and efficiently recover from losses at the point of packet loss. This is due mainly to the efficient distribution of loss recovery burden between the source and receivers; where the source handles losses that occur on the links close to it, and receivers take care from those occurring on the links close to them. Whereas, the receiver-initiated class attributes the responsibility of detection of losses to receivers regardless of the link on which losses occur, thus involving inefficient distribution of loss recovery burden.

The usage of additional services in a router-assisted approach for caching packets and supplying repairs, aggregating flow in feedback and limiting retransmission scoping, addresses the issues of scalability and fairness that arise on large scale. It helps to isolate the domains of loss and, thereby, reduce global retransmissions.

We used the delivery delay, the bandwidth usage, and the buffer requirements as the performance metrics of interest. These metrics are used to study the impacts of different parameters such as the packet loss rate and the multicast group size on the performance of combined class approaches and the received-initiated class alone. We found that combining classes significantly reduced the network bandwidth usage, minimized the buffer requirements at routers, and improved the delivery delay. Based on our numerical results, we argue that combining classes achieves a reliability gain with respect to multicast group size and loss rate compared to a simple receiver-initiated scheme. The performance gains increase as the size of the network and the loss rate increase making the combination of classes more scalable with respect to these parameters.

We showed in this paper that AMRHy adapts perfectly to unreliable environments and can easily migrate to wireless environments as opposed to protocols based on the receiver-initiated class alone.

References

[1] Astuto N., Mendonça M., Nguyen x., Obraczka K., and Turletti T., "A Survey of Software Defined Networking: Past, Present, and Future of Programmable Networks," *in Communications Survey and Tutorials*, vol. 16, no. 3, pp. 1617-1634, 2014.

[2] Coras F., Domingo J., Maino F., Forinacci D., and Cabellos-Aparicio A., "Lcast: Software Defined Inter-Domain Multicast," *Computer Networks*, vol. 59, pp.153-170, 2014.

[3] Cui W. and Qian C., "Dual-Structure Data Center Multicast using Software Defined Networking," *in Computing Research Repository (CoRR)*, 2014.

[4] Derdouri L., Pham C., and Benmohammed M., "A Comparative Analysis of Reliable Multicast Protocols in Active Networking Environments," *International Review on Computers and Software (IRECOS)*, vol. 3, no. 3, pp. 265-275, 2008.

[5] Derdouri L., Pham C., and Benmohammed M., "AMRHy: Hybrid Protocol for Reliable Multicast Transport in Active Networking Environment," *International Journal on Computer Science and Network Security (IJCSNS)*, vol. 8, no. 10, pp. 68-78, 2008.

[6] Derdouri L., Pham C., and Benmohammed M., "Analysis of Delay Latency of the Active Reliable Multicast Protocols," *in Proceeding of the International Arab Conference on Information Technology*, Benghazi, 2010.

[7] Derdouri L., Saidouni D., and Benmohammed M., "Reliable Multicast Transport in Active Environment," *in Proceeding of CSIT'06 Conference Computer Science and Information Technology*, Amman, 2006.

[8] Huang L., Hung H., Lin C., and Yang D., "Scalable Steiner Tree for Multicast Communications in Software Defined Networking," *in Computing Research Repository (CoRR)*, 2014.

[9] Hussain S., *Active and Programmable Networks for Adaptive Architectures and Services*, Auerbarch Publications, 2007.

[10] Kalpana G. and Punithavalli M., "Reliable Broadcasting Using Efficient Forward Node Selection for Mobile Ad-Hoc Networks," *The International Arab Journal of Information Technology*, vol. 9, no. 4, pp. 299-305, 2012.

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

[12] Kasera S., Bhattacharyya S., Kurose J., Zabele S., Kiwior D., Keaton M., and Towsley D., "Scalable Fair Reliable Multicast Using Active Services," *IEEE Network*, vol. 14, no. 1, pp. 48-57, 2000.

[13] Lehman L., Garland S., and Tennenhouse D., "Active Reliable Multicast," *in Proceeding of INFCOM'98*, CA, pp. 581-589, 1998.

[14] Levine B. and Garcia-Luna-Aceves J., "A Comparison of Reliable Multicast Protocols," *Multimedia systems*, vol. 6, no. 5, pp. 334-348, 1998.

[15] Lyer A., Kumar P., and Mann V., "Avalanche: Data Center Multicast using Software Defined Networking," *in Proceeding of 6th International Conference of Communication Systems and Networks (COMNETS)*, India, pp. 1-8, 2014.

[16] Maihöfer C. and Rothemel K., "A Delay Analysis of Tree-based Reliable Multicast Protocols," *in Proceeding of the 10th International Conference on Computer Communication and Networks*, Scottsdale, pp. 274-281, 2001.

[17] Maihöfer C., "A Bandwidth Analysis of Reliable Multicast Transport Protocols," *in Proceeding of NGC 2000 on Networked Group Communication*, USA, pp. 15-26, 2000.

[18] Maimour M. and Pham C., "A Throughput Analysis of Reliable Multicast Protocols in Active Networking Environment," *in Proceeding of the 6th IEEE Symposium on Computers and Communications*, Tunisia, pp. 151-158, 2001.

[19] Maimour M. and Pham C., "DyRAM: an Active Reliable Multicast Framework for Data Distribution," *Journal of Cluster Computing*, vol. 7, no. 2, pp. 407-421, 2004.

[20] Network simulator ns-2 http://www.isi.edu/nsnam/ns/, last visited July, 2013.

[21] Pingali S., Towsley D., and Kurose J., "A Comparison of Sender-initiated and Receiver-initiated Reliable Multicast Protocols," *in Proceeding of the 1994 ACM SIGMETRICS Conference on Measurement and Modelling of Computer Systems*, Tennessee, pp. 221-230, 1994.

[22] Yajnik M., Kurose J., and Towsley D., "Packet Loss Correlation in the Mbone Multicast Network," *in Proceeding of Global Telecommunication Conference*, London, pp. 94-99, 1996.

[23] Yeung K. and Wong H., "Caching Policy Design and Cache Allocation in Active Reliable Multicast," *Computer Networks*, vol. 43, no. 2, pp. 177-193, 2003.

[24] Zheng L., Zhang Z., and Parekh R., "Survey Report on PIM-SM Implementations and Deployments," Technical Report, 2013.

Lakhdar Derdouri obtained his PhD degree in computer science from the University Mentouri of Constantine Algeria in 2009. He is currently lecturer at University Mentouri of Constantine. During 2007-2008 he stayed in Laboratory LIUPPA, Pau France. His current research interests are active networking and reliability multicast protocols, wireless mesh networks, wireless sensor networks and network coding.

Congduc Pham obtained his PhD in computer science in July 1997 at LIP6 Laboratory (laboratoire d'informatique de Paris 6), University Pierre and Marie Curie. He also received his habilitation in 2003 from university Claude Bernard, France. He spent one year at University of California, Los Angeles (UCLA) as post-doctoral fellow. From 1998 to 2005, he was associate professor at the University of Lyon; member of the INRIA RESO project in LIP laboratory at the ENS Lyon. He is now professor at the university of Pau and director of LIUPPA laboratory. His research interests include active networking and networking protocols, sensor networks and congestion control. He has published more than 100 papers in international conferences and journals, has been reviewers for numerous of international conferences and magazines, and participates for many conference program committees. He is member of IEEE.

Mohamed Benmohammed received his B.Sc degree from the High School of Computer Science (C.E.R.I), Algiers, Algeria in 1983. And PhD degree in computer science from University of Sidi Belabbes, Algeria in 1997. he is currently Professor at University Mentouri of Constantine. His current research interests are parallel Architecture and high level synthesis.