AMRHy: Hybrid Protocol for a Reliable Multicast Transport in Active Networking Environments

Lakhdar Derdouri[†], Congduc Pham[††], Mohamed Benmohammed[†]

[†]LIRE Laboratory, University of Mentouri, Route Ain el Bey, 25000 Constantine, Algeria
[††]LIUPPA Laboratory, University of Pau et des pays de l'Adour, 64013 Pau Cedex, France

Summary
The capacity gain of active networks has been extensively studied in the reliability of multicast. It has been shown that active networks improve the network reliability by reducing the number of packet retransmission between the source and receivers in lossy networks. However, the existing active reliable multicast protocols are based on the receiver-initiated class that attributes the responsibility of loss recovery to the receivers regardless the links in which the losses occur. This paper proposes a new active reliable multicast protocol where the responsibility of loss recovery is distributed between the source and the receivers by combining classes. The hybrid approach adopted by our protocol takes the advantages of each class, offering efficient mechanisms to solve the scalability problems such as acknowledgement implosion, repair load balancing, recovery isolation or exposure, and the drop to zero with limited capacity receivers. The numerical results show that combining classes significantly improves the throughput and limits the bandwidth needed by control messages. Interestingly, combining classes can outperform the receiver-initiated class depending on the network size and loss probability.

Key words:
Communication protocol, active networks, reliable multicast, multicast IP.

1. Introduction

Earlier research has identified information dissemination such as news updates, stock quote updates, distance learning, video conference and networked virtual environments etc. as important applications that could benefit from multicast capability. The multicast paradigm naturally fits such applications by constructing a routing multicast tree which allows the source to simultaneously reach all the receivers. The well-known strengths of IP multicast are that it saves network bandwidth by duplicating the packets only where it is necessary. However, IP multicast provides only a best effort delivery of data and does not impose any restrictions on the data rate. This gives rise to two problems. First, some of the transmitted data packets from the source may not reach all the receivers. Second, multicast applications sending data at uncontrolled rates can overwhelm network resources, causing congestion problem, and may starve existing unicast applications of available network bandwidth.

A large number of reliable multicast (RM) protocols have been developed for ensuring reliability at the transport layer and also for ensuring a congestion control in the network. A natural approach to recover from packet losses is based on the sender-initiated class in which the source retransmits the lost packet to individual receivers. However, such sender-initiated retransmission does not scale well. Especially in large scale session where the probability that given packet is lost by many receivers is rather high; thus, the source is brought to retransmit the lost packet to each receiver having lost it. Additionally, when a packet is lost in the links close to the source, most receivers would lose that packet. This leads to prohibitive repair traffic which is proportional to the session size. This problem can be prevented if the source is allowed to multicast the repair packet. Nevertheless, many packet losses are not correlated as indicated in studies [16], and different receivers may experience different loss rates. Consequently a repair-locality problem appears where repair traffic is not localized towards the desired receivers, thus causing the exposure problem where receivers receive many unwanted packets in the repair traffic. Sender-initiated class also suffers from the well known implosion problem in which the source is flooded by control traffic. Much of the research in multicasting addresses the repair-locality and implosion problems. The consensus for addressing these problems is to delegate the responsibility of detection and recovery to the receivers. Thus giving rise to new class of protocols named receiver-initiated. The most popular representative of this class is SRM [2], which allows receivers to multicast request packets to entire group. Any receiver with requested packet can multicast it. With perfect synchronization of randomized timers, SRM can effectively solve the implosion problem. Unfortunately the repair-locality problem is not solved by SRM but can be alleviated by local and hierarchical scoping [13]. There are protocols such as RMTP [11], TMTP [17], and LBRM [4] which adopt hierarchical approach to solve the implosion and repair locality problems by imposing a logical tree structure to the multicast session. At the root of each sub-tree a

Manuscript received October 5, 2008
Manuscript revised October 20, 2008

specialized receiver is located to receive requests and perform retransmissions only to its own descendants in the sub-tree. These protocols work without any router support. On the other hand, the protocols such as PGM [14] and LMS [10] need the assistance of routers in order to localize repair packets to region where they can be most effective. Consequently, managing a large number of specialized routers or receivers under network partition or machine failure would create an enormous administrative burden.

Going a step further, and gaining in generality and flexibility, the use of novel approach called active networking in which network nodes: the switches, routers, hubs, bridges, gateways, etc. perform customized computation on packets flowing through them. The network is called an "active network" because new behaviors can be dynamically injected into nodes. The active network model provides a user driven customization of the infrastructure. These concepts can solve the implosion and repair locality problems in effective way by attributing the role of repair locality to the active router close to the losses. Several active reliable multicast protocols have been proposed in the literature such as ARM [7], AER [6], DyRAM [9]. However, all these protocols are based on the receiver-initiated class that attributes the responsibility of loss detection to the receivers regardless the links in which the losses occur. In this paper, we propose a new active reliable multicast protocol in which the responsibility of loss detection is distributed between the source and receivers by combining classes. In this hybrid of classes, the source handles the packet losses that occur in the links close to the source (source links) and the receivers take care of packet losses on the links close to the receivers (tail links). Our goal is to quantify the benefit of combining classes compared to the traditional received-initiated class.

The rest of the paper is organized as follows. Section (2) presents a state-of-the-art on the reliable multicast transport protocols. Section (3) situates the proposed protocol in the classification of RM protocols and its contribution. Section (4) presents the network model and the hypothesis on which our study is based. Sections (5) and (6) establish the comparative study of our protocol with DyRAM [9] in term of throughput and bandwidth respectively. Finally we conclude in section (7).

2. State-of-the-art

Providing a reliable multicast and effective delivery for data dissemination applications on a large scale is a challenge, particularly when the application requires very short delivery latency and high bandwidth. Several RM protocols have been developed to solve the reliability problem in best effort networks such as the Internet where packet losses are far from being rare. The RM protocols, deal with this problem by a trade-off between the forwarding delay and the capacity in bandwidth. These protocols can be classified in two categories: reliable multicast protocols without active services and those based on active services.

2.1 Reliable multicast without active services

Yeung et al. [18] define the taxonomy of reliable multicast protocols in which the protocols are regrouped according the following criteria: in term of sender-initiated or receiver-initiated classes, and in term of hierarchical or timer based approaches. Fig. 1 shows this classification for some illustrative RM protocols.

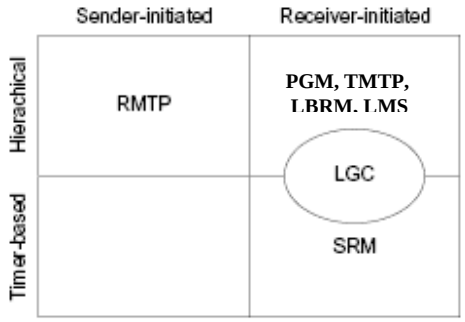

Fig. 1 Classification of reliable multicast protocols.

To ensure reliability, the first RM protocols adopted an end-to-end approach which is based only on the contribution of the source and the receivers. Reliability consists in detecting the losses and making the suitable retransmissions. One must answer the two following questions: who detects the losses and who deals with the retransmission of the lost packets? Two classes of protocols are proposed for this purpose: the sender-initiated and the receiver-initiated.

In the sender-initiated class, only the source is responsible for detection and repair. The task of the receiver is to announce the successful reception of each data packet by sending an ACK. The source detects the loss by monitoring the ACKs. Protocols of this class do not scale well in the presence of a large number of receivers.

In contrast, the receiver-initiated class moves the loss detection responsibility to the receivers. These protocols use NAKs instead of ACKs. Based on this class, a source continues to send new data packets until it receives a NAK from a receiver. Then the requested packet is retransmitted. Since each receiver keeps its own reception state, the per-host state monitoring burden is constant, independent of

group size and of the multicast group management. However, this class still suffers from a NAK implosion problem when an important number of receivers have lost the data packet at the same time.

Generally, the end-to-end approach does not scale well in the presence of a large number of receivers: (a) the source is still in charge of the loss recovery and there is no means to avoid the feedback implosion problem, (b) the exposure problem of the receivers always persists and there is no means to limit retransmission scoping to a subgroup, and (c) the loss recovery latency is too important and there is no means to reduce it.

A solution to these problems is to adopt a local recovery solution based on two approaches: the timer based and hierarchical approaches. The timer based approach uses random timers to solve the NAK implosion problem. When a receiver detects a packet loss, it waits for a random time and then multicasts a repair request. The host close to the point of loss is likely to timeout first and multicasts the request. Other hosts that are also missing the data hear that request and then suppress their own requests. This behaviour prevents NAK implosion. Any host that has a copy of the requested data can answer the request. This approach is particularly robust in scenario with membership or topology changes since it does not depend on an intermediate node to perform NAK suppression and retransmission. SRM [2] is a typical protocol example of this approach. The disadvantage of approach resides in the repair locality problem which can be alleviated by local and hierarchical scoping [13].

The hierarchical approach partitions the multicast delivery tree into subgroups that form a hierarchy rooted at the source. Each subgroup has a leader, designated receivers in [11], log-servers in [4] or designated routers [14, 10] which keeps copies of data packets, collects the feedback from the receivers in the subgroup and retransmits data packets if needed. Feedback implosion is limited because each leader handles a small number of receivers. Limiting the feedback and retransmissions locally saves bandwidth and limits the exposure problem. The recovery latency is reduced as repairs come from representative close to the point of loss. LBRM, RMTP, PGM, TMTP, LMS [4, 11, 14, 17 and 10**]** are typical protocols examples of this approach. The major disadvantage of approach is managing a large number of specialized routers or receivers under network partition or machine failure would create an enormous administrative burden.

The LGC protocol [3] can be regarded as a hybrid protocol of both approaches. It groups receivers into local groups and a group controller in each subgroup is responsible for processing status information from its assigned receivers. Local groups form a tree-like hierarchy, supporting hierarchical data exchange among local groups. Data packet losses are first recovered inside the local groups like in SRM. A group controller requests lost packets from the source or from a higher-level group controller only if no member of its subgroup holds a copy of the lost packet.

2.2 Active reliable multicast protocols

The use of active network concepts, where routers themselves could contribute to enhance the network services by customized functionalities has been proposed in the multicast research community [15]. This approach seems to be a more general solution and more flexible than those based on dedicated and fixed routers such as adopted by the PGM, LMS protocols [14, 10]. The network is called an “active network” because new computations are dynamically injected into the routers, thereby altering the behavior of the network. Packets in an active network can carry fragments of program code in addition to data. In the active reliable multicast protocols, the roles of active services consist mainly in: (a) making the cache of data packets to ensure local loss recovery, (b) performing the aggregation and/or the local NAKs suppression to avoid the feedback implosion problem, (c) limiting the retransmission scoping in the area in which the receivers have lost the data packet, and (d) electing a replier among a subgroup of receivers in order to reduce the load of active routers both in terms of memory and processing tasks.

Several active reliable multicast protocols were developed. We evoke the most cited in the literature, ARM (*Active Reliable Multicast*) [7], AER (*Active Error Recovery*) [6] and DyRAM (*Dynamic Replier Active reliable Multicast*) [9]. All these protocols are based on the receiver-initiated class where the responsibility of loss detection is attributed to the receivers. Here we recall some disadvantages related to this class: (a) high recovery latency that is not acceptable for some real time applications in which not only the reliability is required but also the lower latencies, (b) inefficient distribution of the loss recovery burden between the source and the receivers where losses occurring on the links close to the source will be detected only at the leaves of the multicast tree by the receivers, (c) inefficient management of the active routers cache where there is no means for the active routers to know when they can safely release data from their cache, (d) the risk that a data packet never reaches its destination when the source has a restricted number of buffers in emission, and (e) 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).

3. AMRHy protocol

A solution which remedies to these disadvantages is to combine both sender-initiated and receiver-initiated classes. The alliance of these two classes is translated to the use of positive and negative acknowledgements. The positive acknowledgement has a global meaning: it is used between the receivers and the source to announce the successful reception of a data packet and allows the source to release the emission buffer associated to that data packet and to adjust its emission window. It also permits the active router to: (a) invite its group members having lost the data packet to request it before it will be removed from its cache, (b) inform its group members of the replier address for future repairs without using the active services, (c) inform its group members having correctly received the data packet to remove their ACKs, and (d) release the buffer space occupied by this data packet. The negative acknowledgement is used locally between the active router and its group members to request the lost data packet.

3.1 The contribution of AMRHy

The AMRHy protocol adopts a hybrid solution which combines both approaches as well as both classes. This combination takes the advantages of each class and each approach, offering efficient mechanisms to solve the scalability problems such as acknowledgement implosion, repair load balancing, recovery isolation or exposure, and the drop to zero with limited capacity receivers. The interest of the combination of approaches was already studied and shown in [3] and [6]; however the combination of classes constitutes our principal contribution and its interest will be shown in sections 5 and 6 during the analyzes phase. "**AMRHy**" is the acronym of "***A**ctive **M**ulticast **R**eliable **Hy**brid protocol*". Fig.2 situates the AMRHy in the classification of RM protocols. In this combination of classes, the source handles the losses occurring on the source link (set of point-to-point links that connects the source to the active router) and the receivers take care to those occurring on the tail links (composed of the point-to-point links connecting the active router to each of the receiver). This better distribution of the loss recovery burden between the source and the receivers allows to: (a) eliminate the feedback traffic generated by the receivers when a loss occurs on the source link, (b) avoid the drop-to-zero problem since the first ACK represents the faster receiver in the group, and (c) optimize the use of active routers memory by removing the acknowledged data packets from the cache.

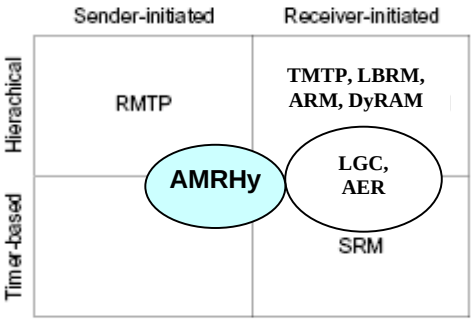

Fig. 2 Position of AMRHy in RM protocols classification.

3.2 The objectives of AMRHy

Combining approaches and classes permits to our protocol to solve the scalability problems and to achieve the design goals of reliable multicast protocols evoked in [6]:
(1) The feedback implosion problem: the suggested protocol avoids this problem by combining both hierarchical and timer based approaches. A first suppression is made according to the SRM principle [2]: when an active router receives the first ACK from one of its descendant and before it forwards it to its ascendant in the multicast tree, it dispatches the ACK towards the others descendants. This permits to the receivers having correctly received the data packets to locally suppress their corresponding ACKs. A second suppression is made by the aggregation of ACKs duplications at a higher level of the multicast tree by the active routers in the same way ARM [7] and DyRAM [9] does: by ignoring the identical ACKs for a well defined period.
(2) The efficient distribution of loss recovery burden: our protocol ensures a better distribution of loss recovery burden between the source and the receivers, by combining sender-initiated and receiver-initiated classes with the contribution of the active routers: (a) the detection of a loss at the source is made when a timer associated to transmitted data packet expires, which means that the loss has occurred on the source link and no receiver has received it, (b) the detection of a loss at the receivers is made when a receiver receives an ACK for a data packet which has not been received, (c) the contribution of the active router in loss recovery is by sending the data packet if it is available in its cache, otherwise the request will be forwarded to the replier (the first receiver having sent an ACK).
(3) Limiting the retransmission scoping: our protocol avoids the exposure problem by maintaining a list structure at the active router like in ARM [7] and DyRAM

[9]. The active router registers in this list structure the address of each receiver announcing the loss of data packet. After a waiting period, the time to know the receivers having lost the data packet, the active router subcasts the data packet to the receivers having requested it.

(4) The minimal router support is guaranteed in our protocol by: (a) the replier election among the receivers; this one ensuring local loss recovery in place of the active router, and (b) the use of positive acknowledgements allows a better management of the active router memory by removing the acknowledged data packet from the cache.

(5) The active services consist essentially in the performance enhancements: our protocol operates correctly without the presence of the active services; it adopts an end-to-end approach in which the loss recovery distribution is shared between the source and the receivers.

3.3 The functional description of AMRHy

AMRHy is a reliable multicast protocol based on active services within routers. These latter are invoked only during multicast sessions, the invocation is done through the multicast tree. Once the services are called, each entity (source, active routers and receivers) behaves in the following way:

The source behaviour

- *When sending a data packet:*
  - Initializes a GD (Guarding Delay) timer, in practice its value is equal to the RTT from the further receiver in the group to the source and will be adjusted each time the source receives an acknowledgement;
  - Stores the data packet in the emission buffer;
  - Sends the data packet to the multicast address subscribed to by all the receivers of the group;
- *When receiving an ACK:*
  - Releases the emission buffer associated to the data packet;
  - Adjusts the source emission window;
- *On timeout of the GD timer:* /* Detection of the loss by the source */
  - Retransmits the data packet to the multicast address;
  - Reinitializes the GD timer;

The receiver behaviour

- *When receiving a data packet:*
  - Initializes a waiting period timer with a random value;
  - Updates the received data packet list;
- *When receiving a repair packet:*
  - Updates the received data packet list;
  - Behaves like that it has sent an ACK to its ascendant ;
- *When receiving an ACK:*
- ***If*** the sequence number does not exist among the received data packet list ***then*** /* Detection of the loss by the receiver */;
  - Initializes the GD timer with a value equal to the RTT between the most distant receiver in the subgroup and its active router;
  - Sends a NAK to the active router;
  - Records the replier address; /* the replier address for future repair*/;

  Else
  - ***If*** waiting period timer is armed ***then*** cancels the waiting period timer ***endif***;
  - Behaves like it has sent an ACK; /* Local suppression of the ACKs */;

  Endif;
- *When receiving a NAK:*
  - /* Only the elected replier can receive a NAK */;
  - Sends the data packet to the receiver having requested it;
- *On timeout of the waiting period timer:* /* The waiting period timer expires before the receiver receives an ACK carrying the same number */
  - Sends an ACK to its ascendant in the multicast tree;
- *On timeout of the GD timer:*
  - Initializes a GD timer with value equal to the RTT between the receiver and the replier;
  - Sends a NAK to the replier;

The active router behaviour

- *When receiving a data packet:*
  - Stores the data packet in the cache;
  - Forwards the data packet to the descendants in this branch of the multicast tree;
- *When receiving a repair packet:*
  - /* This packet is treated in the same manner as the original packet in this branch of the multicast tree*/;
- *When receiving an ACK:*
  If the ACK comes from the descendant ***then***
  - Subcasts the ACK to the other descendants;
  - Initializes waiting period timers with a value equal to the RTT between the active router and its farther descendant in the subgroup;
  - Ignores the ACKs arriving from its descendants during the waiting period; /* the aggregation of the ACKs at active routers */;

  Else /* The ACK comes from the ascendant */
  If the ACK number does not exist in the received data packet list ***then***
  - Sends a NAK to its ascendant in the multicast tree;

- Initializes a GD timer with a value equal to the RTT between the ascendant active router and its farther descendant in the subgroup;
- Records the replier address;

Else /* the data packet was received*/

- Cancels the waiting period timer;
- Behaves like an ACK has been sent to the ascendant; /* local suppression of the ACKs */;

Endif;
Endif;

— *When receiving a NAK:*

- Adds the descendant address into the list structure that contains the receivers addresses having lost the packet;

— *On timeout of the GD timer:*

- Forwards the NAK towards the replier;
- Initializes a GD timer with a value equal to the RTT between the active router and the replier;

— *On timeout of the waiting period timer:*

- Subcasts the repair packet to the receivers having actually requested it;
- Releases the buffer associated to the data packet;
- Sends an ACK to the ascendant,

4. Network model and hypothesis

The network model adopted in our study is similar to the one proposed in [8, 12]. It is based on a multicast tree rooted at the source with receivers at the leaves. Intermediate nodes are the routers (see Fig. 3). The source *S* multicasts the data packets to the *R* receivers which are distributed into *N =R/B* local groups. Each active router is responsible of *B* receivers forming a local group. In our study, we consider that the active routers are placed at strategic point within the network where the losses often occur. These points represent the edge of the backbone for two essential reasons: (a) the backbone is supposed to be reliable. It was shown in [16] that the links where most of the losses occurred are those which are at the edge of the backbone, (b) the backbone is a very high-speed network, if active routers are placed inside it, performance will be degraded. These strategic points will permit to the active routers to intercept any data packet sent by the source towards the receivers of its subgroup for ensuring the local losses recovery. Our study consists in analyzing the needs for AMRHy in terms of throughput and bandwidth and to compare them with those of DyRAM. This later represents receiver-initiated protocols. This comparison will enable us to show the interest of the classes combining. The choice of DyRAM is motivated by the fact that both protocols adopt the same strategy concerning the replier election among the receivers in order to discharge the active router from local loss recovery.

We suppose some hypothesis essential for our analysis:

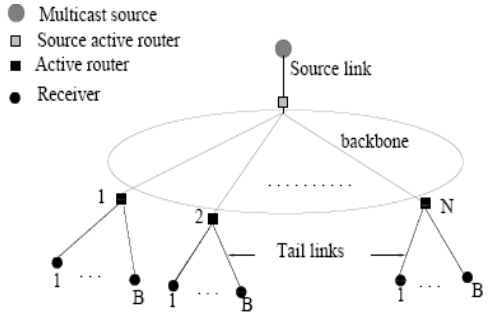

Fig. 3 Network model

- For the loss model, the backbone is considered entirely reliable as mentioned previously, whereas on the other links (source link and tail links) the losses occur with a P_l probability. Therefore, the end-to-end packet loss probability perceived by a receiver is $P=1-(1-P_l)2$. The losses are supposed temporally independent; those relating to the tail links are mutually independent.
- We suppose that a NAK (or ACK) transmitted in unicast follows the same multicast way to be able to benefit from the same active services. This can be realized either by implementing a specific routing service for NAKs (or ACKs), or by saving the address of each active node crossed by the data packet as it was mentioned in **[1].**
-We also suppose that the active router role is summarized in the cache of the data packets, the feedback suppression and the replier election among the local group.

5. Throughput analysis

In this section our analysis is focused on the requirements in term of the processing time for the protocols AMRHy (A) and DyRAM (D). The overall throughput achievable by a protocol depends on the processing time per packet at the various nodes in the multicast tree. For this analysis we use notations similar to those used in [8, 12]. These notations are represented in Table 2. Table 1 summarizes the probabilities distribution and the mean of random variables. For the calculation of the mean of random variable *M*, we apply the following formula:

$$E[M] = 1 + \sum_{m=1}^{\infty} 1 - P[M \leq m]$$

Table 1: Probability distribution and mean

Random variable X	***Probability distribution P[X≤M]***	Mean E[X]
M_s	$1 - P^M$	$\frac{1}{(1-P)}$ OR $E[(M_s-2)^+] = \frac{p^2}{(1-P)}$
M_b	$1- P_l^m$	$\frac{1}{(1-P_l)}$ OR $E[(M_b-2)^+] = \frac{p_l^2}{(1-P_l)}$
M_B	$(1- P_l^m)^B$	$1+\sum_{m=1}^{\infty} 1-(1-P_l^m)^B$

5.1 Analysis of AMRHy

We begin by analyzing the processing requirement at the source: it sends the data packet M_s time until, at least one receiver by local group has correctly received it, treats for that (M_s -1) timeout of GD timer, and consequently receives only one ACK thanks to the local suppression and the aggregation services. The processing time per packet at the source can therefore be written:

$$E[X^A] = E[M_s]E[X_p] + (E[M_s]-1)E[X_t] + E[X_a] \quad (1)$$

Replacing E [Ms] and E [Ms] -1 by their expressions in Table 1 gives:

$$E[X^A] = \frac{1}{1-P}E[X_p] + \frac{P}{1-P}E[X_t] + E[X_a] \quad (2)$$

The processing time at the active routers depends on whether it is on the source link or on the tail link:

– *On a tail link (Ai: i=1.. N).*

An active router (*Ai*) responsible of a local group receives the data packet M_b times from its ascendant and (M_B -M_b) times from the replier of its local group; thus M_B times (because M_b+ [M_B -M_b] = M_B), and sends the same data packet M_B times towards the receivers of its subgroup. Consequently it receives (sends) an ACK and treats for this ACK the timeout of the waiting period timer. It receives (M_s -1) NAKs from each receiver of its subgroup, thus B (M_s -1) from all the receivers; it sends (M_B -1) NAKs towards the replier and treats for these NAKs (M_B -2) timeout of the RT timer. Therefore we obtain the following expression:

$$E[A_i^A] = E[M_B](E[Y_p^a]+E[X_p^a]) + E[X_a^a] + E[Y_a^a] + E[A_t] + (E[M_B]-1)E[Y_n^a] + B(E[M_s]-1)E[X_n^a] + E[(M_B-2)^+]E[A_t] \quad (3)$$

Replacing *E [M_s]* -1 by its expression in Table 1 gives:

$$E[A_i^A] = E[M_B](E[Y_p^a]+E[X_p^a]) + E[X_a^a] + E[Y_a^a] + E[A_t] + (E[M_B]-1)E[Y_n^a] + B(\frac{P}{1-P})E[X_n^a] + E[(M_B-2)^+]E[A_t] \quad (4)$$

– *On the source link A_s.*

The active router (A_s) on the source link receives the data packet from the source M_s times with probability (1-P_l) and sends (1-P_l) M_s packets towards its descendants. It receives only one ACK from one receiver of its subgroup, subcasts the ACK towards the others descendant, treats for that the timeout of the WP timer and sends to the source only one ACK. Therefore the following expression can be written:

$$E[A_s^A] = (1-P_l)E[M_s](E[Y_p^a]+E[X_p^a]) + E[Y_a^a] + E[A_t] + E[X_a^a] \quad (5)$$

Replacing *E [M_B]* by its expression in Table 1 gives:

$$E[A_s^A] = \frac{1}{1-P_l}(E[Y_p^a]+E[X_p^a]) + E[Y_a^a] + E[X_a^a] + E[A_t] \quad (6)$$

If we analyse the processing time at the receiver, it sends (Ms -1) NAKs, treats (MB -2) timeout of the RT timer until the data packet is correctly received; then thanks to the subcast service it receives the data packet only once, consequently it sends or receives only one ACK. If now the receiver is elected as a replier, it sends the data packet as many time as it receives NAK packets. This number is estimated to be: $\frac{E[M_B]-1}{B}$.

Therefore we have the following expression:

$$E[Y^A] = E[Y_p] + E[Y_a] + (E[M_s]-1)E[Y_n] + E[(M_s-2)^+]E[Y_t] + (\frac{E[M_B]-1}{B})(E[X_p]+E[X_n]) \quad (7)$$

Replacing *E [M_B -1]* and E *[M_B -2]* by their expressions in Table 1 gives:

$$E[Y^A] = E[Y_p] + E[Y_a] + \frac{P}{1-P}E[Y_n] + \frac{P^2}{1-P}E[Y_t] + (\frac{E[M_B]-1}{B})(E[X_p]+E[X_n]) \quad (8)$$

5.2 Analysis of DyRAM

Again, the analysis begins by defining the processing requirements at the source: it sends a data packet M_s times until, at least one receiver per local group has correctly received it, and thus receives (M_s -1) NAKs. The processing time per packet at the source can therefore be written:

$$E[X^D] = E[M_s]E[X_p] + (E[M_s]-1)E[X_n] \quad (9)$$

Replacing *E [M_B]* and E *[M_B -1]* by their values expressions in Table 1 gives:

$$E[X^D] = \frac{1}{1-P}E[X_p] + \frac{P}{1-P}E[X_n] \quad (10)$$

Table 2: Notations used in analytical evaluation of throughput

Analytical model variable X	Meaning
X^A, X^D	The total processing time per packet at the source for protocols **A** and **D**.
A_i^A , A_i^D	The total processing time per packet at an active router A_i(i=1..N) associated to local group for protocols **A** and **D**.
A_s^A , A_s^D	The total processing time per packet at the active A_s associated to the source for protocols **A** and **D**.
Y^A, Y^D	The total processing time per packet at a receiver for protocols **A** and **D**.
X_p, X_n, X_a	The processing time for sending data packet and to receive a NAK (ACK).
Y_p, Y_n, Y_a	The processing time for receiving data packet and to send a NAK (ACK).
X_p^a , X_n^a , X_a^a	The processing time for sending a data packet and to receive a NAK (ACK) by an active router respectively.
Y_p^a , Y_n^a , Y_a^a	The processing time for receiving a data packet and to send a NAK (ACK) by an active router respectively.
X_t	The processing time for treating a timeout by the source.
Y_t	The processing time for treating a timeout by a receiver.
A_t	The processing time for treating a timeout by an active router.
M_s	The number of transmissions necessary for sending a data packet from the source until it is correctly received by one receiver.
M_b, M_B	The number of transmissions necessary for sending a data packet from an active router until it is correctly received by one or all receivers of its local group.

The processing time at the active routers depends on whether it is on the source link or on the tail link:

– *On a tail links (Ai: i=1.. N).*

An active router (*Ai*) responsible of a local group receives the data packet M_b times from its ascendant and (M_B -M_b) times from the replier of its local group; thus M_B times and sends the same data packet M_B time towards the receivers of its local group. Consequently it sends (M_b -1) NAKs towards the source and treat for these NAKs (M_b -2) timeout of RT timer, it receives (M_s -1) NAKs from each receiver of its local group; thus $B^*(M_s$ -1) from all the receivers, it sends (M_B -1) NAKs towards the replier and treats for these NAKs (M_B -2) timeout of the RT timer and elects a replier during a DTD period (the value of this time is not represented in table 2, we suppose that it is equal to the value of A_t). Therefore the following expression can be written:

$$\begin{aligned} E[A_i^D] &= E[M_B](E[Y_p^a]+E[X_p^a])+(E[M_b]-1)E[Y_n^a] \\ &+E[(M_b-2)^+]E[A_t]+(E[M_B]-1)E[Y_n^a] \\ &+E[(M_B-2)^+]E[A_t]+E[A_{dtd}^a]+B^*(E[M_s]-1)E[X_n^a] \end{aligned} \quad (11)$$

Replacing *(E[Ms]-1), (E[Mb-1])* and *(E[Mb-1])* by their expressions in Table 1 gives:

$$\begin{aligned} E[A_i^D] &= E[M_B](E[Y_p^a]+E[X_p^a])+\frac{P_l}{1-P_l}E[Y_n^a]+\frac{P_l^2}{1-P_l}E[A_t] \\ &+E[M_B-1)E[Y_n^a]+E[(M_B-2)^+]E[A_t] \\ &+B^*(\frac{P}{1-P})E[X_n^a]+E[A_{dtd}^a] \end{aligned} \quad (12)$$

– *On the source link A_s*

The active router (A_s) on the source link receives the data packet from the source M_s times with the probability (1-P_l) and sends (1-P_l) M_s towards its descendants. It sends for the same data packet (M_s -1) NAKs towards the source and treats for these NAKs (M_s -2) timeout of RT timer. It receives on average $N^*(E[M_s]$-1) NAKs from its descendants. Therefore the following expression can be written:

$$\begin{aligned} E[A_s^D] &= (1-P_l)E[M_s](E[Y_p^a]+E[X_p^a])+(E[M_s]-1)E[Y_n^a] \\ &+E[(M_s-2)^+]E[A_t]+N^*(E[M_s-1)E[X_n^a]) \end{aligned} \quad (13)$$

Replacing $E[M_s]$, ($E[M_s$-1]) and ($E[M_s$-2]) by their expressions respectively in Table 1 gives:

$$E[A_s^D]=\frac{1}{1-P_l}(E[Y_p^a]+E[X_p^a])+\frac{P}{1-P}(E[Y_n^a]+N^*E[X_n^a])+\frac{P^2}{1-P}E[A_t] \quad (14)$$

The processing time requirement at a receiver: each receiver in protocol **D** as in protocol **A** receives only once data packet thanks to the subcast service, it sends (M_s -1) NAKs, and treats (M_B -2) timeout of the RT timer until the data packet is correctly received. If now the receiver is elected as a replier, it sends the data packet as many times it receives the NAK**.** This number is estimated to be: $\frac{E[M_B]-1}{B}$.

Therefore the following expression can be written:

$$\begin{aligned} E[Y^D] &= E[Y_p]+(E[M_s]-1)E[Y_n]+E[(M_s-2)^+]E[Y_t] \\ &+\frac{E[M_B]-1}{B}(E[X_p]+E[X_n]) \end{aligned} \quad (15)$$

Replacing ($E[M_s$-1]) and ($E[M_s$-2]) by their expressions in Table 1 gives :

$$\begin{aligned} E[Y^D] &= E[Y_p]+\frac{P}{1-P}E[Y_n]+\frac{P^2}{1-P}E[Y_t] \\ &+\frac{E[M_B]-1}{B}(E[X_p]+E[X_n]) \end{aligned} \quad (16)$$

5.3 Numerical results

For the numerical evaluation of the overall throughput the following values are taken:

$E[X_p]=E[Y_p]=E[X_p^a]=E[Y_p^a]=500\ \mu\text{sec}$

$E[X_n]=E[Y_n]=E[X_a]=E[Y_a]=E[X_n^a]=E[Y_n^a]=E[X_a^a]=E[Y_a^a]=85\mu\text{sec}$

$E[X_t]=E[Y_t]=2E[A_1]=32\ \mu\text{sec}$

These values are those experimentally measured in **[9].**

The throughput Λ_x^w achieved by node x under the protocol $w \in \{\mathbf{A}, \mathbf{D}\}$ is calculated by the formula:

$$\Lambda_x^w = 1/E[x^w],\ x \in \{X, Y, A_1 \ldots\ldots A_N, A_S\} \quad (17)$$

The overall throughput Λ^w achieved by the protocol w is then given by:

$$\Lambda^w = Min\ (\Lambda_x^w) \quad (18)$$

In order to know which nodes yield the minimum throughput, Fig. 4 plots the throughput achieved by each node in **A** according to the number of local groups. We can note that: (a) the minimal throughput is the one introduced by the active routers at tail links A_i, this is due mainly to the load put on them to ensure local recovery, (b) the receivers have a maximum throughput thanks to the subcast service which avoids the exposure problem, (c) the throughput achieved by each node remains constant regardless the number of local groups, (d) the overall throughput achieved by **A** is thus achieved by an active router at the tail link.

Fig. 5 plots the throughput achieved by each node in **D** according to the number of local groups. In the same manner that in the previous figure we can note that: (a) the minimal throughput is introduced at the beginning by the active routers at the tail links A_i, and then when increasing the number of local groups (approximately 100) the active router on the source link A_s becomes the bottleneck, (b) the receivers have a maximum throughput thanks to the subcast service which avoids the exposure problem, (c) the throughput achieved by the other nodes remains constant regardless the number of local groups, (d) the overall throughput achieved by **D** is the one achieved at the beginning by an active router at tail link A_i, and then the one achieved by the active router at source link when the number of local groups increases.

Fig. 6 presents a comparison of **A** and **D** in term of the overall throughput. We note that at the beginning the throughput is almost the same for both protocols with a light advantage for **D**. When the number of local groups is increased (approximately N> 95), the throughput of **A** remains constant but that of **D** decrease in a significant manner. This result can be interpreted by the fact that in **A**, the ACKs benefit from local suppression and aggregation services resulting from the combination of both hierarchical and timer based approaches. In **D**, the NAKs benefit only from one service of aggregation of the hierarchical approach. In addition, combining classes allow a better distribution of loss recovery burden between the source and the receivers with the contribution of the active routers. Combining classes also allow the active router in the source link to remove the overhead of the tail links losses. However, in **D**, the active router in the source link will be confronted to the overhead of all the losses occurring on the tail links. Furthermore, the ACKs generate a feedback flow only once to announce the good reception of the data packet, whereas a feedback flow generated by the NAKs is repeated until the data packet has correctly been received by all the receivers when the loss occurs on the source link.

Fig.7 shows that by increasing the loss probability the overall throughput decreases especially for D. This result encourages us to plot the curves of the overall throughput of both protocols according to the loss probability.

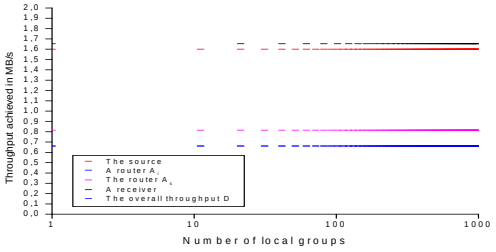

Fig. 4 Throughput achieved by different nodes in **A** (B=10, P=0,05)

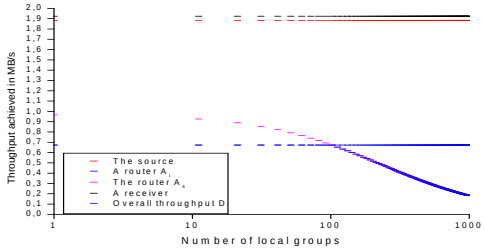

Fig. 5 Throughput achieved by different nodes in **D** (B=10, P=0,05)

Fig. 8 shows that the overall throughput achieved by **D** is higher than the one of **A** when P varies in the interval [0, 0.05]. Consequently, if the P is higher than 0.05, the throughput achieved by **D** becomes lower than the one achieved by **A**. This result confirms that combining classes is better for adapting to unreliable environments than the receiver-initiated class alone.

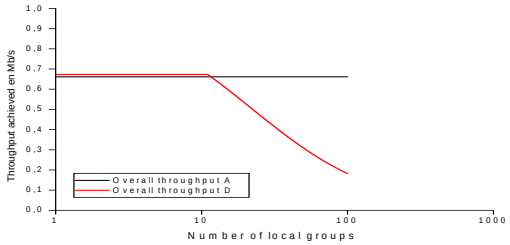

Fig. 6 Throughput achieved by **A** and **D** (B=10, P=0,05)

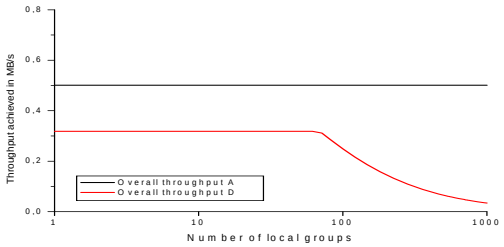

Fig. 7 Throughput achieved by **A** and **D** (B=10, P=0,25)

Fig. 9 shows that for an important number of local groups (N=500), the difference in decrease speed becomes definitely clear. **A** is better than **D** in term of throughput when the P is higher than (0.05). That 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 the link on which the losses occur. This result confirms that combining classes is more scalable in unreliable environments than the receiver-initiated class alone.

6. Bandwidth analysis

This section the analysis focuses on the requirements in term of consumed bandwidth by AMRHy (A) and DyRAM (D) protocols. To make this analysis similar notations are used to those in [5]. These notations are represented in table 3. To analyze the performance of a protocol in term of consumed bandwidth we consider three types of link: the source link, the backbone link and the tail link. The total consumed bandwidth will thus be given by the following expression:

$$B^w = E[B_s^w] + N * E[B_b^w] + R * E[B_t^w] \quad (19)$$

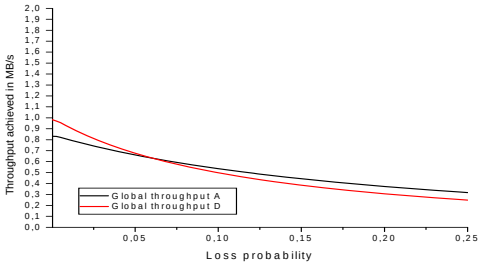

Fig. 8 Throughput achieved by **A** and **D** (B=10, N=100)

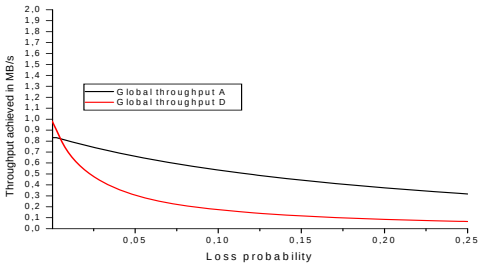

Fig. 9 Throughput achieved by **A** and **D** (B=10, N=500)

To determine the various terms of this expression, we need to find the number of packets which crosses these three links (the source link, the backbone link and the tail link) so that a data packet transmitted by the source is correctly received by all the receivers.

6.1 Analysis of AMRHy

The bandwidth consumed on the various links by **A** is:

Source link:

$$E[B_s^A] = E[M_s]E[B_p] + E[B_a] \quad (20)$$

Backbone link:

$$E[B_b^A] = (1 - P_l)E[M_s]E[B_p] + E[B_a] \quad (21)$$

Tail link:

$$E[B_t^A] = E[B_p] + (E[M_s] - 1)E[B_n] + E[B_a] + (\frac{E[M_B] - 1}{B})(E[B_p] + E[B_n]) \quad (22)$$

6.2 Analysis of DyRAM

The bandwidth consumed on the various links by **D** is:

Source link:

$$E[B_s^D] = E[M_s]E[B_p] + (E[M_s] - 1)E[B_n] \quad (23)$$

Backbone link:

$$E[B_b^D] = (1 - P_l)E[M_s]E[B_p] + (E[M_s] - 1)E[B_n] \quad (24)$$

Tail link:

$$E[B_t^D] = E[B_p] + (E[M_s] - 1)E[B_n] + \left(\frac{E[M_B] - 1}{B}\right)(E[B_p] + E[B_n]) \quad (25)$$

Table 3: Notations used in analytical evaluation of bandwidth

Analytical model variable X	Meaning
B^w	The total bandwidth consumed by protocol w.
B_s^w	The bandwidth consumed on the source links by protocol w.
B_b^w	The bandwidth consumed on the backbone links by protocol w.
B_t^w	The bandwidth consumed on the tail links by protocol w.
B_p , B_n , B_a	The bandwidth consumed by a data packet and a NAK or an ACK.

6.3 Numerical results

For the evaluation of the consumed bandwidth, we take the same values as those taken in [5]: E [B_p] = 1024 and E [B_n] = E [B_a] =32.

Fig. 10 shows that with a loss probability of P=0.01 the consumed bandwidth by **D** is lower than that consumed by **A**.

Fig. 11 shows that by increasing the loss probability to P=0.05 the consumed bandwidth by both **A** and **D** becomes identical.

Fig. 12 shows that by increasing further the loss probability (P=0.25), the consumed bandwidth by **D** becomes higher than that consumed by **A.** This result can be explained by the fact that the combination of classes reduces the feedback flow of the NAKs when the loss occurs on the source link and consequently reduces the consumption of bandwidth.

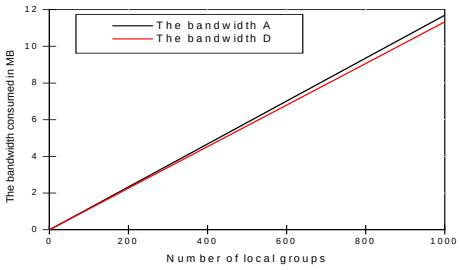

Fig. 10 Consumed bandwidth in **A** and **D** (B=10, P=0.01)

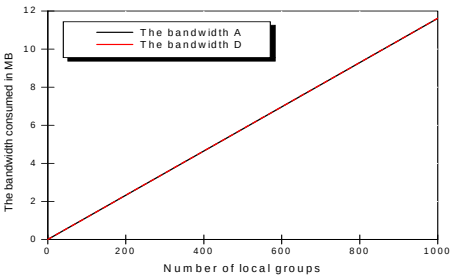

Fig. 11 Consumed bandwidth in **A** and **D** (B=10, P=0.05)

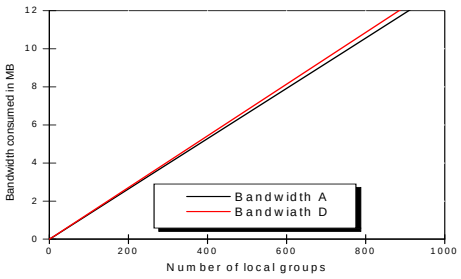

Fig. 12 Consumed bandwidth in **A** and **D** (B=10, P=0.25)

7. Conclusion

We have proposed a novel reliable multicast protocol based on active networking concepts. In the receiver-initiated class protocols, the responsibility of loss detection is attributed to receivers regardless of links on which the losses occurred causing an inefficient distribution of the loss recovery burden between the sender and the receivers. In our protocol, by combining sender-initiated and receiver-initiated classes, the responsibility for detection of losses is efficiently distributed between the source and receivers. In this hybrid approach, the link on which a loss occurred is taken into account and the source handles losses occurring on its close links (source links) while the receivers take care from those occurring on their close links (tail links). The hybrid approach adopted by our protocol takes the advantages of each class, offering efficient mechanisms to solve the scalability problems that emerge at a large scale such as acknowledgement implosion, repair load balancing, recovery isolation or exposure, and the drop to zero with limited capacity receivers. Using analytical analysis, we demonstrate the performance gains in combining classes in terms of protocol throughput and bandwidth consumption. 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.

For future works we plan to extend our performance analysis to study the loss recovery latency and its complex interactions on the cache size at the active routers.

References

[1] L. Derdouri, D. E. Saidouni, M. Benmohammed. Reliable Multicast Transport in Active environment. In proceeding of Conference Computer Science and Information Technology CSIT'06 (Page: 411 Year of Publication: 2006 ISBN: 9957-8592-x).

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

[3] M. Hofmann. Enabling Group Communication in Global Network. In proceeding Global networking'97 (Page: 321 Year of Publication: 1997 ISBN-10: 9051993455).

[4] H. W. Holbrook, S. K. Singhal, D. R. Cheritoy. Log-based Receiver Reliable Multicast for Distributed Interactive Simulation. In ACM SIGCOMM Computer Communication Review, Vol. 25, n. 4, pp. 328-341, 1995.

[5] 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 17 th IEEE Computers and Communications INFOCOM'98 (Page: 988 Year of Publication 1998 ISBN: 0-7803-4383-2).

[6] S K. Kasera, and all. Scalable Fair Reliable Multicast Using Active Services. In proceedings of DARPA Active Networks Conference and Exposition (DANCE'02) (Page: 333 Year of Publication: 2002 ISBN: 3-540-41179-8).

[7] L. H. Lehman, S. J. Garland, D. L. Tennenhouse. Active Reliable Multicast. In proceedings of IEEE INFCOM'98, San Francisco, CA, (Page: 581 Year of Publication 1998 ISBN: 0-7803-4383-2).

[8] M. Maimour and C. Pham. A throughput Analysis of Reliable Multicast Protocols in Active Networking Environment. In proceeding of the sixth IEEE symposium on Computers and Communications (Page: 151 Year of Publication; 2003 ISBN-10: 0769511775).

[9] M. Maimour, C. Pham. DyRAM: an Active Multicast Framework for Data Distribution. In Journal of cluster Computing, Vol. 7, n. 2, pp. 163-176, 2004.

[10] C. Papadopoulos, G. Parulkar, nand G. Varghese. An Error Control Scheme for Large Scale Multicast Applications. In proceeding of the IEEE INFOCOM (Page: 1188 Year of Publication: 1998 ISBN: 0-7803-4383-2).

[11] S. Paul, K. Sabnani, J. C. Lin, S. Bhattacharyya. Reliable Multicast Transport Protocol (RMPT). In IEEE Journal on Selected Areas in Communications, Vol. 15, n. 3, pp. 207-421, 1997.

[12] S. Pingali, D. Towsley, and J. F. Kurose. A Comparison of Sender-initiated and Receiver-initiated Reliable Multicast Protocols. In AM SIGMETRICS94 (Page: 221 Year of Publication: 1994 ISBN: 0-89791-659-x).

[13] P. Sharma, D. Estrin, S. Floyd, and L. Zhang. Scalable Session Messages in SRM using Self-configuration. In USC Technical report, July 1998.

[14] T. Speakmanand al. PGM Reliable Multicast Protocol. In IEEE Network, Vol. 17, n. 1, pp. 16-22, 2003.

[15] D. L. Tennehouse, J. M. Sincoskie, D. J. Wetherall, and G. J. Winden. A Survey of Active Network Research. In IEEE Communication Magazine, Vol. 35, n. 1, pp. 80-86, 1997.

[16] M. Yajnaik, J. Kurose, and D. Towsley. Packet Loss Correlation in the MBone Multicast Network. In proceedings of Global Telecommunication Conference (Page: 94 Year of Publication: 96 ISBN: 0-7803-3336-5).

[17] R. Yavatkar, J. Griffoen, and M. Sudan. A Reliable Dissemination Protocol for Interactive Collaborative Applications. In proceedings of ACM Multimedia (Page: 333 Year of Publication: 1995 ISBN: 0-89791-791-0).

[18] K. L. Yeung, H. T. Wong. Caching Policy Design and Cache Allocation in Active Reliable Multicast. In Computer Networks, Vol. 43, n. 2, pp. 177-193, 2003.

Lakhdar Derdouri received the B.S. and Magister degrees in computer science from University Mentouri of Constantine Algeria in 1985 and 1998, respectively. 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 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 40 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 was born in Constantine, Algeria on December 26, 1959. He 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.