Performance Analysis of Loss Recovery Latency in Reliable Multicast Protocols using Active Parity Encoded Services

Lakhdar Derdouri[1] and Congduc PHAM[2]

[1]*LIRE Laboratory, University of Mentouri, Route Ain El Bey 25017 Constantine, Algeria*
[2]*LIUPPA Laboratory, University of PAU et des pays de l'Adour, 64013 Pau Cedex, France*
derdouril@yahoo.fr, congduc.pham@univ-pau.fr

Abstract

Providing an efficient and reliable multicast for data dissemination applications on a large scale is a challenge, especially when the applications require a very short delivery delay and high throughput. The combination of a local recovery approach based on active services with those using FEC/ARQ gives rise to a new class of reliable multicast protocols called APES "Active Parity Encoding Services". This paper carries out a comparative study between reliable multicast protocols belonging to this class in terms of loss recovery latency. We use the simulation to study the impact of coding/decoding time and loss rate on the performance of this class of protocols. Our numerical results show that the approach Get-Repairs Store-Repairs, besides the reduction in storage space at the active router, provides a substantial gain in terms of loss recovery latency and thus contributes to improve the real time reliable multicast.

Keywords: *Reliable Multicast, FEC codes, Active services, Latency*

1. Introduction

Multicast provides an economic way of disseminating information from a sender to a group of receivers located at different networks locations. It is an efficient means to support applications such as videoconferencing, distributed gaming, distance learning, IP-TV and VoD (*video-on-demand*), *etc*. Besides the effectiveness of routing at the network layer, most of these applications not only require reliable and efficient multicast networking services, but also a small data delivery delays. However, heterogeneous networks and variable traffics make the design of reliable multicast protocols more problematic.

Several protocols have been proposed to solve the problem of reliability in networks in which delivery is known as *best effort* such as the Internet where packet losses are frequent. RM (*Reliable Multicast*) protocols address this problem by imposing a compromise between the routing delay and the capacity in bandwidth.

Traditional approaches operate in an end-to-end way with an automatic retransmission of lost packets from the source (*Automatic Repeat Request*, ARQ). Several mechanisms such as filtering control messages or restricting retransmissions help to prevent overloading the source and receivers while allowing the scalability in the presence of important groups [4]. An alternative approach uses techniques of correction codes to produce repair packets in order to answer to possible losses of data packets at the receivers (*Forward Error Correction*, FEC). Receivers, in this approach, are required to perform additional coding/decoding operations to use or reproduce the original data [1, 2, 15, 9, 11].

A certain category of RM protocols uses the active networking technology where routers themselves could contribute to enhance the network services by performing customized computations on the packets flowing through them [5]. The active network model has the ability to provide a very general and flexible framework for customizing network functionalities in order to gracefully handle heterogeneity and dynamics. This category of protocols is based on the local recovery of packet losses by using RS (*Repair Services*, RS). In this paper we will use the terms "repair services" or "active services" interchangeably. These repair services contribute mainly in solving the implosion and repair locality problems in a more effective way, by attributing the role of repair to the router close to the loss location. The active routers are placed at strategic points within the network; they intercept data packets and store them in their caches to locally ensure the recovery of losses while processing the negative acknowledgments (NAKs) sent by receivers for these losses [6, 8, 16, 18, 3].

FEC/ARQ and RS approaches reduce the requirements in terms of bandwidth and the loss recovery latency of RM protocols. The coupling of these two approaches gives rise to a new class of reliable multicast protocols called (*Active Parity Encoding Services*, APES) [7, 10]. Several variants of APES protocols have emerged in order to improve the performance of reliable multicast. Among these variants, two APES protocols have been proposed in the literature: BRSR (*Built-Repairs-Store Repairs*) and GRSR (*Get-Repairs-Store Repairs*) [14]. These protocols were compared with an oldest variant of APES protocols: SDBR (*Store-Data Built-Repairs*) [17]. A comparative study highlighted the improvements in terms of reduction in storage space at the repair server and in terms of bandwidth, provided by these two APES protocols [14]. This comparative study showed that BRSR and GRSR protocols provide a considerable reduction in storage space at the repair server while maintaining bandwidth consumption close to that observed in SDBR protocol [14].

In this paper, we propose to extend the comparative study to include the loss recovery latency as a metric for evaluating the performance of this class of protocols. Our study shows that the loss recovery latency grows linearly according to the coding/decoding time and according to the loss rate. For a high coding/decoding time, GRSR presents the best performance. However, for a lower coding/decoding time, SDBR is the protocol which minimizes the transmission time of a data block.

The remainder of the paper is organized as follows: Section 2 gives a brief description of the APES protocols (SDBR, BRSR and GRSR). Section 3 presents the network model on which our study will be based. Assumptions of the comparative study are presented in Section 4. Section 5 deals with the comparative study of the APES protocols in terms of loss recovery latency. Section 6 concludes this paper and sets directions for future works.

2. Description of the variants of APES protocols

In this section we present three variants of APES protocols: SDBR, BRSR and GRSR. BRSR and GRSR protocols have been proposed with the objective to improve the performance of reliable multicast protocols in terms of bandwidth and storage capacity at repair servers [14, 17]. These protocols were compared with an oldest variant, SDBR, to highlight the achieved improvements. Globally, these three protocols differ in how and when they generate repair packets at the repair server, and in the choice of packets to be stored in the cache of the repair server. In this section, we discuss the tasks assigned to repair servers by each protocol to ensure a reliable transmission of data blocks for receivers in their repair domains.

2.1. SDBR (Store-Data Built-Repairs)

In this protocol, the source sends a combination of data packets and repair packets in blocks of k packets to all concerned repair servers. Once a repair server has received a block of k packets, it decodes this block to restore the original data packets and store them in its cache. Then repair server sends the block of k packets to the receivers.

At the receivers, the received source packets are decoded by a FEC decoder to restore the original data packets. If a receiver needs additional FEC packets in order to repair the losses of a block, it sends a NAK to its repair server indicating the number of additional repair packets needed. Upon the reception of a NAK, a repair server generates new requested repair packets with a FEC coder, and sends them in a restricted way (subcast) to receivers of its domain. Since FEC packets are built on purpose, the receiver can use the obtained repair packets to repair any loss occurred in the block of k packets.

2.2. BRSR (Built-Repairs Store-Repairs)

BRSR is another variant of the APES protocols. The repair server defines in advance a number b of repair packets to be generated for each block. In this case, the data packets received by the repair server are fetched into a FEC coder to generate b repair packets that will be stored in its cache (this type **of** coder allows on-the-fly coding) [12, 13]. The value of b is between 0 and k, ($0 \leq b \leq k$), where k is the number of packets per block. These b packets occupy the space of the buffer cache upon the reception of the first source packet and cannot be used for repair until the k^{th} source packet is introduced into the FEC coder by the repair server. In the best case, when all receivers lose less than b source packets, the b repair packets stored at the repair server are sufficient to recover any loss reported by the receivers.

The repair server subcasts the repair packets requested by the receivers in its domain. If a receiver loses more than b source packets, then the repair server must obtain additional repair packets by sending towards the source a NAK indicating the number of missing FEC packets. The repair server is in fact unable to generate by itself further FEC packets as the data packets are not available in its cache.

After receiving the additional packets, it sends them in a restricted way to the receivers in its domain. This protocol assumes that the source is capable to send additional repair packets at any time and without loss. The innovations introduced by this protocol are:

- Instead of the cache of data packets, only the repair packets are stored in the cache of the repair servers.
- It is not necessary to generate new repair packets at each loss; it is sufficient to send repair packets stored in the cache of repair servers.

2.3. GRSR (Get-Repairs Store-Repairs)

As in BRSR, GRSR defines a number b of repair packets generated by block. However, unlike BRSR, repair servers do not perform coding/decoding operations.

They get the *b* required repair packets from the source and then store them in their caches upon their arrival.

The repair packets are constructed at the source and then transmitted directly after the block of *k* packets for which they were generated. Once the *b* repair packets are received and stored at a repair server, this latter behaves like in BRSR for the rest of the transmission.

3. Network Model

A commonly used model for evaluating multicast protocols is to have a multicast tree that admits the source as the root and the repair servers as intermediate nodes with receivers as leaves (see Figure 1). The receivers are distributed according a topology in *m* repair domains $D_1 \dots D_m$ [14]. The size of a repair domain is defined by the number of receivers in this domain $|D_i|$. In reality, the receivers can either be LANs containing one or more receiving applications or simply other repair servers; these latter are being responsible for repair domains in lower levels. In the context of active networking, we consider that the active routers are placed at strategic locations within the network where the losses often occur. These locations are usually located at the edge of the backbone for two essential reasons:

- The backbone is supposed to be reliable: [19] showed that the links where most of losses occurred are those located at network's edge.
- The backbone is a very high-speed network and adding complex processing functions inside the backbone will certainly degrade its performance.

This organization allows repair servers to intercept any packet sent from the source towards the receivers in its domain to ensure the local recovery of losses. They are also able to diffuse in a restricted way packets to their repair domain so that the data reach only the receivers of their domain.

The source gathers the data packets in blocks of size *k*, and fetches them into a FEC coder for generating repair packets. The number of repairs that can be generated for a block is finite and large enough to ensure the recovery of possible losses.

The source is supposed to be able to reproduce indefinitely additional repairs if this is required. Repair packets, generated for a block of *k* packets, are supposed to belong to this block. The *k* packets of a block (combination of data and repair) are called source packets. Any entity (receiver or repair server) with a FEC decoder is able to restore the original data packets regardless to the combination of *k* received packets (data and repair) in this block. Protocols based on FEC code must ensure that each receiver receives at least *k* packets per block to ensure reliability. The fact that different data packets can be lost and that the same set of repair packets can be used to repair losses reduces the number of transmissions required for loss recovery. This reduction was observed in an end-to-end basis [12]. Repair servers aim to ensure that each receiver belonging to its repair domain has received at least *k* distinct packets per block. To accomplish that, each protocol derived from the APES class determines which tasks are performed by a repair server.

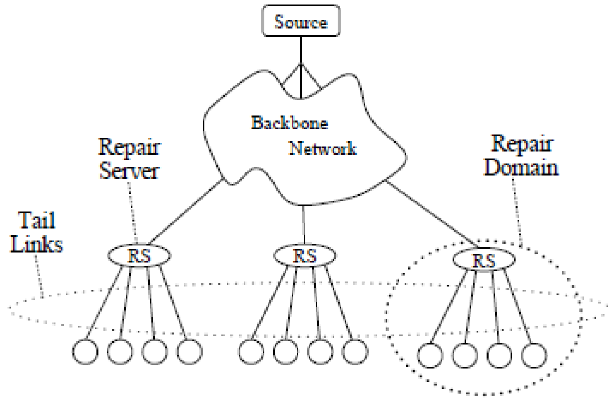


Figure 1. Network Model

4. Hypothesis of Comparative Study

The study compares three variants of APES protocols: SDBR, BRSR and GRSR in terms of loss recovery latency of a block of k packets sent by a source until its correct reception by all receivers. This will enable us to determine, among the three variants of protocols, which variant is the most adapted to real time reliable multicast. Our study is based on a network model with a single hierarchical level of repair servers but can be easily extended to a network model with several hierarchical levels. In this case, each repair server at level i provides its repair services to a set of repair servers of level $i+1$ ($1 \leq i \leq n$ where n is the depth of hierarchy). The benefit of choosing the one hierarchy level network model is that each repair domain can be analyzed separately from the other. Thus, we base our study on the topology of Figure 2 and we have the following assumptions:

- Let r be the number of receivers in the repair domain, this number indicates the domain size.
- The links between the repair server and the receivers have identical properties; in particular they all have a packet loss rate p.
- The probability of losing packets between the source and the repair server is negligible.
- The repair server assigns to k source packets a sequence number varying from 1 to k. Repair packets have sequence numbers greater than k.
- A receiver can detect a loss after having received the source packets. A receiver, having lost m out of the k source packets requests the retransmission of ($k+1$, ..., $k+m$) repair packets from the repair server. In practice, a receiver requesting m repairs packets and having lost only one of them can actually use a repair packet numbered $k+m'$ such as $m' \leq m$ in place of the lost packets.

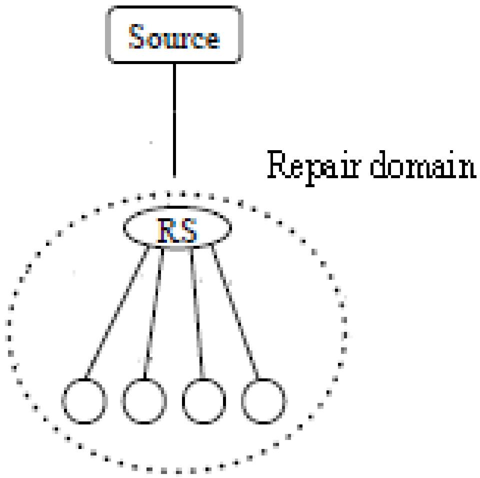


Figure 2. Single hierarchical level model

5. Analysis of loss recovery latency

In this section we determine the time needed to transmit, in a reliable way, a block of k packets from the source until its correct reception by all receivers using the three protocols under consideration. Thus we define T_{SDBR}, T_{BRSR} and T_{GRSR} as the overall time needed to send a block of k packets using SDBR, BRSR and GRSR protocols respectively. The evaluation is performed with the ns2 simulator. We set a coding/decoding time (which will be identical for the three protocols) at each entity. Before we present the simulation results, the various operations of coding/decoding performed by each variant of these protocols are summarized in Table 1.

Our first study consists to determine for the three protocols whether the choice of the block size has an influence on the transmission time. Figure 3 represents the transmission time T_{SDBR}, T_{BRSR} and T_{GRSR} according to the block size k by using SDBR, BRSR and GRSR protocols respectively (the domain size is $r = 8$, and the loss rate is $p = 0.05$). The curve showed that the overall transmission time is a linear function of the block size. Since the losses will be detected only when the k source packets will be received by the receiver and, given that the transfer time of a block, in a reliable way, depends on the block size, the smaller the block size, the shorter the time required to detect and to repair the losses. For block sizes smaller than 40 packets, all three protocols require a similar transmission time. For larger block sizes, SDBR requires less time compared to the other protocols because it performs a local recovery regardless of the number of losses that have occurred at the receivers. BRSR requires more time because, besides the time to perform various operations of FEC coding/ decoding for local recovery of losses, it also requires additional time to inform the source in case it needs additional repairs from it.

Based on the result showed in Figure 4, we can see that the transmission time of SDBR, BRSR and GRSR protocols increases linearly according to the FEC coding time. For a small coding time, BRSR is better than GRSR in terms of transmission time. Beyond a certain value of the coding time, GRSR becomes better than BRSR and SDBR because the source in GRSR sends additional repair packets right after the k source packets, thus avoiding the of coding/decoding overheads at the repair server.

Table 1. Comparison of APES protocols

	SDBR	**BRSR**	**GRSR**
Repair server	• Decoding the k packets of a block to extract original data packets. • Coding of original data packets stored in the buffer to generate the asked repair packets if an additional request for repair arrives.	• Decoding the k packets of a block to generate b repair packets and to be stored in its cache.	• No operation.
Receiver	• Decoding of k packets of the block to restore the original data packets.	• Decoding of k packets of the block to restore the original data packets.	• Decoding of k packets of the block to restore the original data packets.

Figure 5 presents the transmission time T_{SDBR} and T_{GRSR} of SDBR and GRSR protocols according to the loss rate p. In this case we set the domain size r = 8, the coding/decoding t_{code} = 0.05 and the block size k = 10.

We can see that the loss rate plays a significant role in determining the contribution of SDBR and GRSR protocols in terms of transmission time of a block. The transmission time of the two protocols increases with the loss rate at the receivers. More losses in the network generate more coding/decoding operations and more access to the repair server.

Figure 6 presents the transmission time ratio between GRSR and SDBR according to the coding/decoding time. The graph clearly shows that as the coding time becomes close to the time required for transferring data from the repair server to the receivers the more GRSR the transmission time diverges from SDBR the transmission time.

GRSR transmission time is significantly reduced compared to that of the SDBR protocol. The increase in the coding time affects more clearly the transmission delay of SDBR protocol than that of GRSR protocol because the repair server of SDBR protocol performs the coding task each time a loss is announced whereas the repair server does not perform any coding task under GRSR.

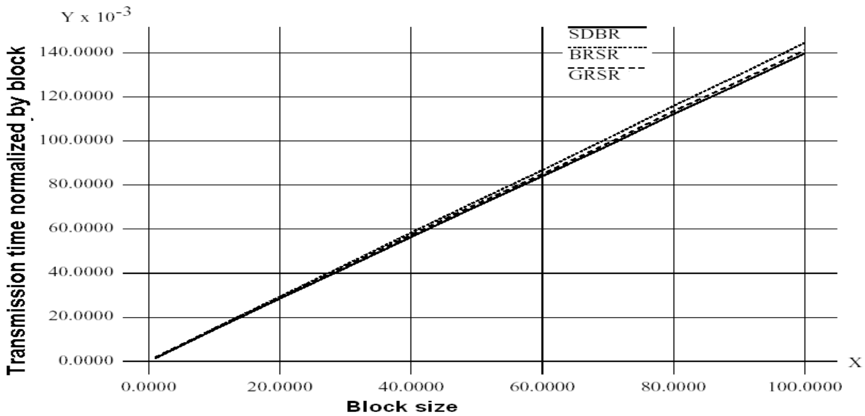


Figure 3. Transmission time according to the block size

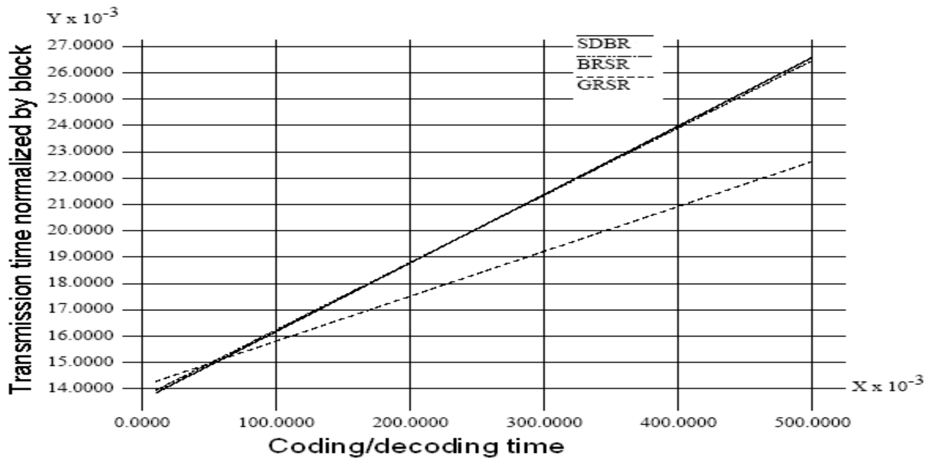


Figure 4. Transmission time according to coding/decoding time

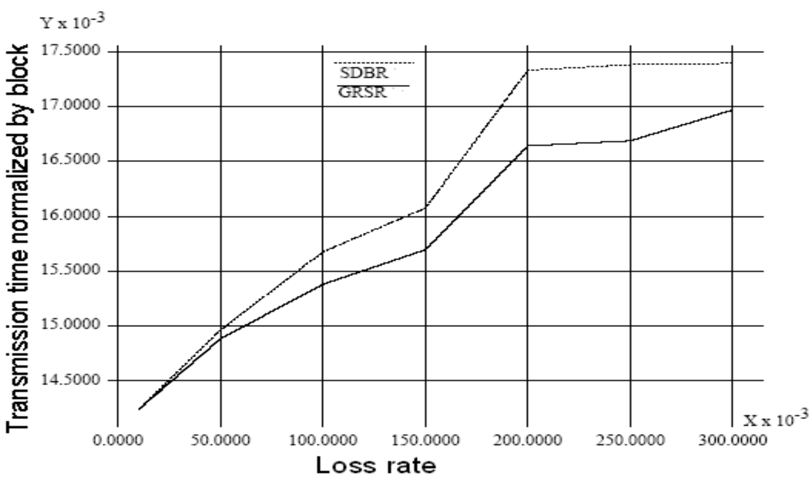


Figure 5. Transmission time according to the loss rate

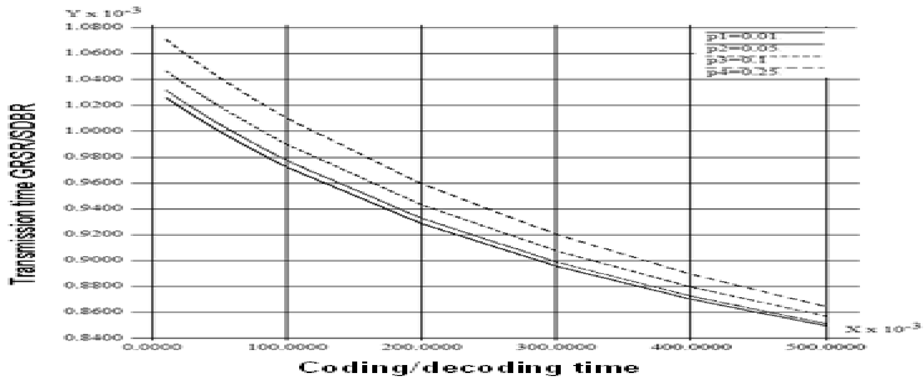


Figure 6. Ratio GRSR/SDBR in term of transmission time according to the coding time for various loss rates

6. Conclusion

In this paper 3 reliable multicast protocols using Active Parity Encoded Services are compared: BRSR, GRSR and SDBR. Our analysis of the loss recovery latency shows that the loss rate and coding/decoding time determine whether a protocol can be a candidate for real time reliable multicast or not. Our study showed that GRSR protocol, in addition to the reduction in storage space at the repair server, provides a substantial gain in terms of loss recovery latency and thus contributes to improve real time reliable multicast transfers. For high coding time, GRSR presents the best performance, whereas, for a lower coding time SDBR is the protocol which minimizes the transmission time of data block.

In this paper we have intentionally studied a simple network model with one hierarchical level to facilitate the results interpretation. Moreover, the packet loss rate was assumed to be homogeneous throughput the network. In future works, we will aim to generalize the results of our study to a network model with *n* hierarchical levels where receivers can have different loss rates. Another research direction for us is to study the relationship between the buffer cache size at repair server and the recovery latency.

References

[1] J. W. Byers, M. Luby and M. Mitzenmacher, "Accessing Multiple Mirror Sites in Parallel: using Tornado Codes to Speed up Downloads", Proceedings of IEEE INFOCOM'99, New York, NY, USA, **(1999)** March, pp. 275-283.

[2] J. Byers, M. Luby, M. Mitzenmacher and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data", Proceedings of SIGCOMM'98, New York, NY, USA, **(1998)** September, pp. 56-67.

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

[4] S. Floyd, V. Jacobson, C. G. Liu, S. Mc Canne and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application level Framing". IEEE/ACM Transactions on Networking, vol. 5, no. 6, **(1997)**, pp. 784.

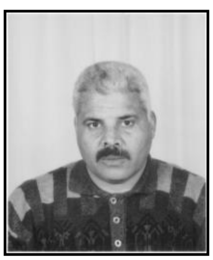
[5] E. Hossain and K. K. Leung, "Wireless Mesh Networks Architectures and protocols", Springer Science + Business Media, LCC, **(2008)**.

[6] S. Kasera, J. Kurose and D. Towsley, "A Comparison of Server-based and Receiver-based Local Recovery Approaches for Scalable Reliable Multicast", Proceedings of IEEE INFOCOM'98, San Francisco, CA, USA, **(1998)** March, pp. 988-995.

[7] R. Kermode, "Scoped Hybrid Automatic Repeat Request with Forward Error Correction (SHARQFEC)", Proceedings of SIGCOMM'98, New York, NY, USA, **(1998)** September, pp. 278-289.

[8] L. H. Lehman, S. J. Garland and D. L. Tennenhouse, "Active Reliable Multicast", Proceedings of IEEE INFOCOM'98, San Francisco, CA, USA, **(1998)** March, pp. 581-589.
[9] [LIC 99] D. Li and D. R. Cheriton, "Evaluating the Utility of FEC with Reliable Multicast", Proceedings of the International Conference on Network Protocols, Atlanta, GA, USA, **(1999)** October, pp. 97-105.
[10] J. Nonnenmacher, M. Lacher, M. Jung, E. Biersack and G. Carle, "How bad is reliable multicast without local recovery", Proceedings of INFOCOM'98, San Francisco, CA, USA, **(1998)** March, pp. 972-979.
[11] J. Nonnenmacher, E. Bircsack and D. Towsley, "Parity-Based Loss Recovery for reliable Multicast Transmission", IEEE/ACM Transactions on Networking, vol. 6, no. 4, **(1998)** August, pp. 349-361.
[12] J. Nonnenmacher, E. Bircsack and D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmission", ACM SIGCOMM '97, Cannes, France, **(1997)** September, pp. 289-300.
[13] L. Rizzo, "Effective Erasure codes for Reliable Computer communication Protocols", ACMSIGCOMM Computer Communication Review, vol. 27, no. 2, **(1997)** April, pp. 24-36.
[14] D. Rubenstein, S. Kasera, D. Towsley and J. Kurose, "Improving Reliable Multicast Using Active Parity Encoding Services (APES)", Computer Networks-COMPUT NETW, vol. 24, no. 1, **(2004)**, pp. 63-78.
[15] D. Rubenstein, J. Kurose and D. Towsley, "A Study of Proactive Hybrid FEC/ARQ and Scalable Feedback Techniques for reliable Real Time Multicast", Computer Communications, vol. 24, no. 5-6, **(2001)** March, pp. 563-574.
[16] D. Rubenstein, N. F. Maxemchuk and D. Shur, "A centralized Tree Based Approach to Network Repair Service for Multicast Streaming Media", Proceedings of IEEE NOSSDAV'00, Chapel Hill, NC,USA, **(2000)** June, pp. 247-256.
[17] D. Rubenstein, S. Kasera, D. Towsley and J. Kurose, "Improving Reliable Multicast Using Active Parity Encoding Services (APES)", Technical Report 98-79, Computer Science Department, University of Massachusetts at Amherst, (**1998**) July.
[18] K. Sripanidkulchai, A. C. Meyers and H. Zhang, "A third-party value-Added Network Service Approach to Reliable Multicast", Proceedings of ACM SIGMETRICS'99, Atlanta, GA, USA, **(1999)** May, pp. 166-177.
[19] M. Yajnik, J. Kurose and D. Towsley, "Packet Loss Correlation in the Mbone Multicast Network", Proceeding of Global Telecommunication Conference, London, England, **(1996)** November, pp. 94-99.

Authors



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 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.