An Active Reliable Multicast Framework for the Grids*

M. Maimour and C. Pham

Laboratoire RESAM, Université Lyon 1
ENS, Bât. LR5, 46 allée d'Italie
69364 Lyon Cedex 07, France Congduc.Pham@ens-lyon.fr

Abstract. Computational Grids are foreseen to be one of the most critical yet challenging technologies to meet the exponentially growing demands for high-performance computing in a large variety of scientific disciplines. Most of these grid applications imply multiple participants and, in many cases, make an intensive usage of data distribution and collective operations. In this paper, we propose a multicast framework consisting of an active reliable protocol with specialized active services located at the edges of the core network for providing low-latency and low-overhead multicast transfers on computational grid.

1 Introduction

Computational Grids are foreseen to be one of the most critical yet challenging technologies to meet the exponentially growing demands for high-performance computing in a large variety of scientific disciplines. Most of these grid applications imply multiple participants and, in many cases, make an intensive usage of data distribution and collective operations. In the past few years, many grid software environments for gaining access to very large distributed computing resources have been made available (e.g. Condor [1], Globus [2], Legion [3] to name a few) and efficient data distribution is usually a key functionality for these environments to reduce the end-to-end latency.

These multi-point communications could be gracefully and efficiently handled by multicast protocols, that is the process of sending every single packet to multiple destinations, provided that these protocols are well-designed to suit the grid requirements. Motivations behind multicast are to handle one-to-many communications in a wide-area network with the lowest network and end-system overheads. In contrast to best-effort multicast, that typically tolerates some data losses and is more suited for real-time audio or video for instance, reliable multicast requires that all packets are safely delivered to the destinations. Desirable features of reliable multicast include, in addition to reliability, low end-to-end delays, high throughput and scalability. These characteristics fit perfectly the need of the grid computing and distributed computing communities.

* This work has been supported in part by the french ACI Grid program and by a grant from ANVAR-EZUS Lyon.

Meeting the objectives of reliable multicast is not an easy task. In the past, there have been a number of propositions for reliable multicast protocols relying on complex exchanges of feedback messages (ACK or NACK) [4–7]. These protocols usually take the end-to-end solution to perform loss recoveries. Most of them fall into one of the following classes: sender-initiated, receiver-initiated and receiver-initiated with local recoveries. In sender-initiated protocols, the sender is responsible for both the loss detection and the recovery (XTP [4]). These protocols usually do not scale well to a large number of receivers due to the ACK implosion problem at the source. Receiver-initiated protocols move the loss detection responsibility to the receivers. They use NACKs instead of ACKs. However they still suffer from the NACK implosion problem when a large number of receivers have subscribed to the multicast session. In receiver-initiated protocols with local recovery, the retransmission of a lost packet can be performed by any receiver (SRM [5]) in the neighborhood or by a designated receiver in a hierarchical structure (RMTP [6], TMTP [7], LMS [8], PGM [9]). All of the above schemes do not provide exact solutions to all the loss recovery problems. This is mainly due to the lack of topology information at the end hosts.

In this paper, we propose a multicast framework consisting of an active reliable protocol with specialized active services located at the edges of the core network (adding complex processing functions inside the core network will certainly slow down the packet forwarding functions) for providing low-latency and low-overhead on computational grid. We assume that the computing resources are distributed across an Internet-based network with a high-speed backbone network in the core (typically the one provided by the telecommunication companies) and several lower-speed (up to 1Gbits/s) access networks at the edge, with respect to the throughput range found in the backbone.

The rest of the paper is organized as follows. Section 2 presents the active reliable multicast concepts. Section 3 describes the generic protocols used in this paper to compare the various active mechanisms involved in reliable multicasting. Then Section 4 presents the performance study and Section 5 concludes.

2 Active networking and the grids

In active networking [10], routers themselves play an active role by executing application-dependent functions on incoming packets. Recently, the use of active network concepts where routers could contribute to enhance the network services by customized (sometimes high-level) functionalities has been proposed in many research areas including multicast protocols [11, 12] and distributed interactive simulations [13]. These results can be very beneficial to the grid community on numerous ways: (*i*) program and data remote submissions, (*ii*) information services and naming services, (*iii*) data replication and caching, (*iv*) collective operations and clock synchronization for distributed applications, (*v*) large databases replication...

There are many difficulties, however, for deploying in a large scale an active networking infrastructure. Security and performance are two main difficulties that are usually raised by anti-active networking people. However, active networking has the ability to provide a very general and flexible framework for customizing network functionalities in order to gracefully handle heterogeneity and dynamicity, key points in a computational grid. Therefore we believe it is still worth proposing advanced mechanisms that would globally improve the quality of service on a distributed system such as the grid.

Now, if we look more precisely at multicasting, active services contribute mainly on feedback implosion problems, retransmission scoping and cache of data. Resulting active reliable multicast protocols open new perspectives for achieving high throughput and low latency on wide-area networks:

- the cache of data packets allows for local recoveries of loss packets and reduces the recovery latency.
- the suppression of NACKs reduces the NACK implosion problem.
- the subcast of repair packets to a set of receivers limits both the retransmission scope and the bandwidth usage.

For instance, ARM (Active Reliable Multicast) [11] and AER (Active Error Recovery) [12] are two protocols that use a *best-effort* cache of data packets to permit local recoveries. ARM adopts a *global* suppression strategy: a receiver experiencing a packet loss sends immediately a NACK to the source. Active services in routers then consist in the aggregation of the multiple NACKs. In contrast, AER uses a *local* suppression strategy inspired from the one used by SRM and based on local timers at the receivers. In addition, an active router in ARM would send the repair packet only to the set of receivers that have sent a NACK packet (subcast). In AER, the active router simply multicasts the repair packet to all its associated receivers.

3 The DyRAM Framework

DyRAM is a reliable multicast protocol suite with a recovery strategy based on a tree structure constructed on a per-packet basis with the assistance of routers [14]. The protocol uses a NACK-based scheme with receiver-based local recoveries where receivers are responsible for both the loss detection and the retransmission of repair packets. Routers play an active role in DyRAM which consists in the following active services:

1. the early detection of packet losses and the emission of the NACKs.
2. the NACK suppression of duplicate (from end-hosts) NACKs in order to limit the NACK implosion problem.
3. the subcast of the repair packets only to the relevant set of receivers that have experienced a loss. This helps to limit the scope of the repairs to the affected subtree.
4. the replier election which consists in choosing a link as a replier one to perform local recoveries from the receivers.

Compared to existing active reliable multicast protocols such as ARM and AER, DyRAM has been designed with the following motivations in mind: (*i*) to minimize active routers load to make them supporting more sessions (mainly in unloading them from the cache of data) and (*ii*) reduce the recovery latency for enabling distributed applications on the grid. Avoiding cache in routers is performed by a replier-based local recovery strategy. The election of the replier is done dynamically on a per-packet basis by the appropriate active service (number 4) instead of an approximate solution based on timers as in SRM, or a complex Designed Local Receiver discovery as in PGM, both being non-active solutions. Reducing the latency recovery is performed by several mechanisms (actually all 4 services contribute directly or indirectly to reduce the recovery latency; for instance active service number 2 avoids NACK implosion at the source therefore reducing the retransmission time from the source) and especially by the early loss packet detection active service (active service number 1).

3.1 The NACK-based strategy

In DyRAM the receivers are responsible for detecting, requesting and in most cases, retransmitting a lost data packet. A receiver detects a loss by sequence gaps and upon the detection of a loss (or on a receive-timer expiration for delayed packets), a receiver immediately sends a NACK toward the source and sets a timer. Since NACKs and repairs may also be lost, a receiver will re-send a similar NACK when the requested repair has not been received within the timeout interval. Practically, the timeout must be set to at least the estimated round trip time (RTT) to the source. Upon reception of a NACK packet, the source sends the repair packet to the multicast address.

3.2 Active services in DyRAM

At the active routers side, the loss packet detection, the NACK suppression, the subcast and the replier election services can be implemented simply by maintaining information about the received packets and NACKs. This set of information is uniquely identified by the multicast address. For each received NACK, the router creates or simply updates an existing NACK state (NS) structure which has a limited life time.

- *seq*: the sequence number of the requested packet,
- *time*: the time when the last valid NACK for this packet has been received,
- *rank*: the rank of the last received NACK. The last valid NACK has rank 1; the next received one, which is not valid, has rank 2 and so forth ...
- *subList*: a subcast list that contains the list of links (downstream or upstream) on which NACKs for this packet have arrived.

Packet losses detection In general the repair latency can be reduced if the lost packet could be requested as soon as possible. DyRAM realizes this functionality by enabling routers to detect losses and therefore to generate NACKs to be sent to the source. An active router would detect a loss when a gap occurs in the data packet sequence. On a loss detection, the router would immediately generate a NACK packet toward the source. If the router has already sent a similar NACK for a lost packet then it would not send a NACK for a given amount of time. This "*discard delay*" is set at least to the RTT between this router and the source. During this amount of time, all NACK packets received for this data packet from the downstream links are ignored. This loss detection service can be implemented without adding any additional soft state at the routers.

NACK suppression On receipt of a NACK packet, a router would look for a corresponding NS structure. If such a structure exists, the router concludes that at least one similar NACK has already been processed otherwise a new NS structure will be created for this NACK. In the former case the router additionally checks if this NACK is valid, and if so, the router will forward it on the elected replier link (how this election is performed is described later on). Otherwise this NACK will serve to properly update the NS structure (rank and subcast list) and is dropped afterward. We use the rank information in some cases to be sure that all downstream receivers have sent a NACK packet.

Subcast functionality The subcast list in the NS structure is an important component for the subcast functionality. This list contains the set of links (downstream or upstream) from which a NACK has been received. When a data packet arrives at an active router it will simply be forwarded on all the downstream links if it is an original transmission. If the data packet is a repair packet the router searches for a corresponding NS structure and will send the repair on all the links that appear in the subcast list. An NS is created as soon as a valid NACK arrived for which no previous NS was created, and can be replaced only on receipt of the corresponding repair.

Replier Election Local recoveries, when possible, are performed by elected receiver (replier) that have correctly received the loss packet. On reception of a valid NACK, the router initializes a timer noted DTD (Delay To Decide) in order to collect during this time window as much information as possible about the links affected by a loss (updates of the subcast list). On expiration of the DTD timer, the router is able to choose a replier link among those that are not in the subcast list. This link may end up to be the upstream one if all the downstream links appear to be in the subcast list. In an attempt to avoid for the overloading of a particular downstream link, the router always try to choose a different link from the previously elected one (if available) by respecting a ring order among them, thus realizing when possible a load balance.

3.3 Architectural design and deployment

While not relying on active services, LMS [8] and PGM [9]) had previously proposed router-assisted solutions and [17] has investigated the incremental deployment of LMS-based routers. In our case, we push for a fully active solution in the hierarchy of routers of lower-speed access networks. In principle, the nearest router to the backbone is a good location for installing the active services: the cache of packets can serve for several local computing resources for instance.

For the realization of our active approach, the *programmable switch approach* which maintains the existing packet format will be used. Programs are injected separately from the processing of messages. Initially, and using a reliable channel, the source injects the required set of services by sending them to the multicast address. In our case this consists in two services, a data packet service and a NACK service. Afterwards, the source begins the multicast of the data packets. When an active router receives a packet, it first looks for the appropriate service deduced from the packet header.

To dynamically handle the multicast group topology changes, active routers must be able to add or remove active services. An active router that leaves the multicast session has simply to remove the associated services. However when an active router joins a multicast tree, it has to download the required services. This can be achieved by requesting the services from the closest active router which has already received them. If there is no such active router then the services need to be sent from the source.

All the active routers are assumed to perform at least NACK suppressions and the subcast functionality. The active router which is located at the sender's side just before the core network is called the *source router*. A preliminary study, detailed in [15], has shown that the loss detection service is only beneficial if the loss detection capable-router is close enough to the source. Consequently the source router is the best candidate to perform the loss detection service in addition to the two previous services. The other active routers perform the replier election service as seen in the previous section.

4 Simulations

4.1 Network model and metrics

We implemented a simulation model of DyRAM (in the PARSEC language developed at UCLA [16]) and evaluated its performances on a network model. The network model considers one source that multicasts data packets to R receivers through a packet network composed of a fast core network and several slower edge access networks. We will call *source link* the set of point-to-point links and traditional routers that connects the source to the core network. Similarly, a *tail link* is composed of point-to-point links and routers connecting a receiver to the core network (see Fig. 1). Each active router A_i is responsible of B receivers noted $R_{i1}, \cdots, R_{iB}$ forming a local group. We assume that there are l_b backbone links between the source and every active router A_i.

To evaluate our protocol M_S is defined as the number of retransmissions from the source per packet and gives an idea on the load at the source and we use BW which is the average bandwidth consumed per link to represent the load at the network level.

$$M_S = \frac{Number\ of\ retransmissions}{Number\ of\ sent\ packets}, BW = \frac{Bw_{NACK} + BW_{Data}}{N.B + l_b.(N+1)}$$

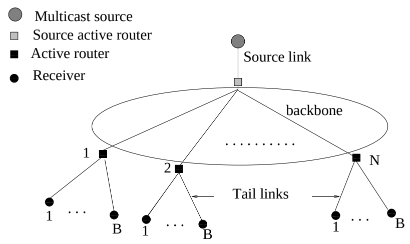

Fig. 1. Network Model.

where BW_{NACK} and BW_{Data} are respectively the bandwidth consumed by the NACK packets and the data packets during the multicast session:

$$BW_{NACK} = \sum_{i=1}^{NackNb} l_{NACK_i}, BW_{Data} = \sum_{i=1}^{DataNb} l_{DP_i}$$

l_{NACK_i} and l_{DP_i} are respectively the number of links crossed by the ith NACK packet and the ith data packet. The third metric is the completion time per packet which is the required time to successfully receive the packet by all the receivers (can be seen as the latency).

4.2 Simulation results

For all the simulations, we set $l_b = 55$. A NACK and a data packet are considered to be of 32 and 1024 bytes respectively (most of data packets on an internet-grid would have size less than 512 bytes and probably 10KBytes at the maximum with GigaEthernet). All simulation model values are normalized to the NACK transmission time (e.g. the time required to send or receive a NACK is set to 1, for a data packet this time is set to 32). For the processing overheads at the routers, we assume that both NACKs and data packets are processed in 32 time units. In our simulation models, we have taken into consideration the fact that the repairs may also be lost.

Local recovery from the receivers Figure 2a and 2b plot for DyRAM the number of retransmissions (M_S) from the source as a function of number the receivers and the completion time. These results have been obtained from simulations of a multicast session with 48 receivers distributed among 12 local groups. The curves show that having local recoveries decreases the load at the source as the number of receivers increases. This is especially true for high loss rates. Putting the recovery process in the receivers requires at least 2 receivers per active router otherwise local recoveries can not be realized. Therefore the local group size (B) is an important parameter. In order to study the impact of B, simulations are performed with the 48 receivers distributed among groups of different sizes and figure 2c shows how much bandwidth (in ratio) can be saved

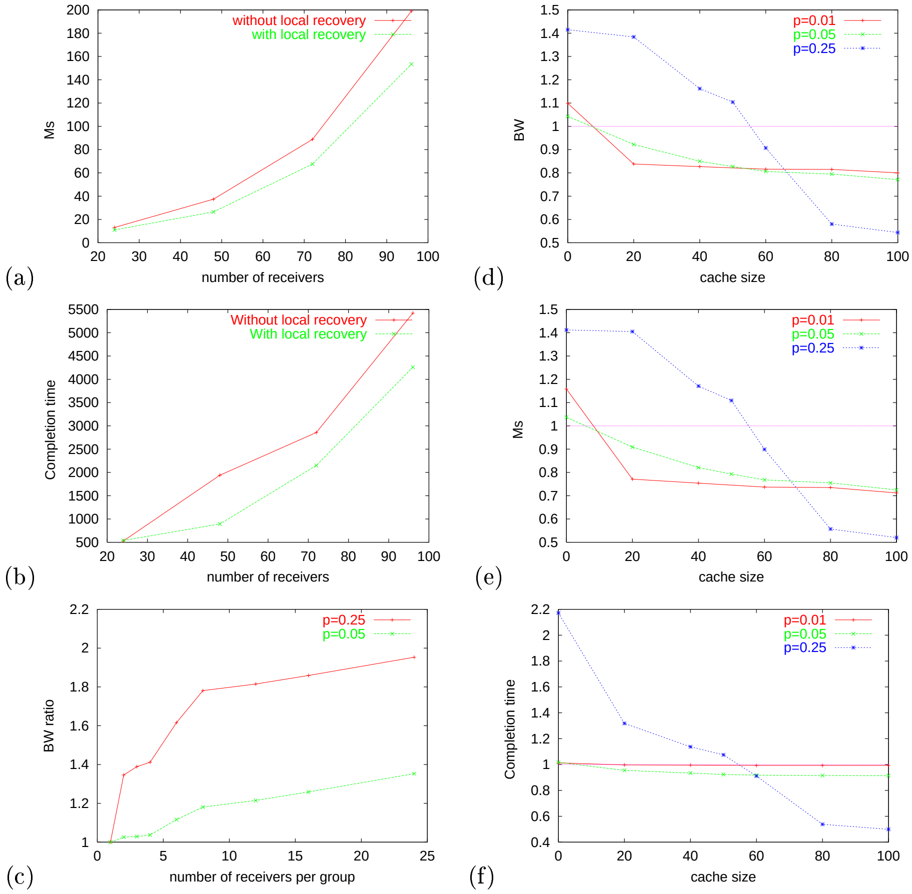

Fig. 2. (a) Load at the source, $p = 0.25$. (b) Completion time, $p = 0.25$. (c) Consumed bandwidth ratio. DyRAM vs. ARM: (d) Consumed bandwidth, (e) Load at the source, (f) Completion time.

with the local recovery mechanism. As the number of receivers per group increases, the consumed bandwidth is reduced by larger ratios for large loss rates. In fact, when the group size is larger it is more likely that the members of the group can recover from each other. For instance, figure 2c shows that with only 6 receivers per group and local recovery we can achieve a gain of 80%.

Early loss packet detection The next experiments show how the early loss detection service could decrease the delay of recovery. To do so, two cases noted DyRAM- and DyRAM+ are simulated. DyRAM- has no loss detection services whereas DyRAM+ benefits from the loss detection service in the source router. Figure 3 plots the recovery delay (normalized to the RTT) for the 2 cases as a function of the number of receivers. In general, the loss detection service allows

DyRAM+ to complete the transfer faster than DyRAM-. For $p = 0.25$ and 96 receivers for instance, DyRAM+ can be 4 times faster.

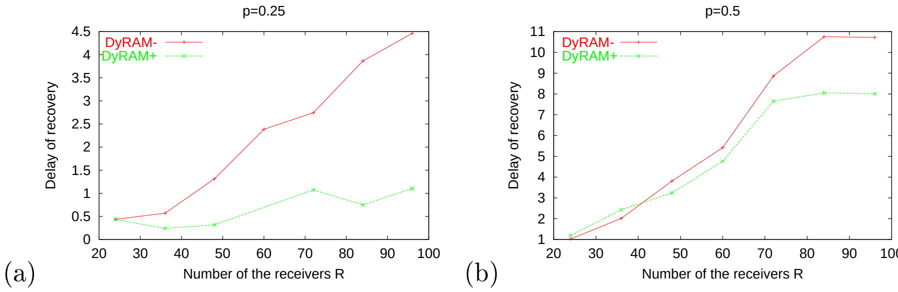

Fig. 3. The normalized recovery delay with (a) $p = 0.25$ and (b) $p = 0.5$

DyRAM vs. ARM In this section, we compare our approach to ARM. Figures 2d, 2e and 2f show for different loss rates the ratio of the consumed bandwidth (BW), the ratio of the load at the source in term of number of retransmissions and the ratio of the completion time. We took into consideration the percentage of the available cache at the ARM routers. Without any cache at the routers, DyRAM always performs better than ARM. When ARM benefits from caching at the routers, it performs better only for very low loss rates or large cache size at the routers. However for large loss rates (e.g. 25%), ARM requires more than 50% of cache at the routers to perform better than DyRAM. Our approach is more promising because even without any caching overhead at the routers DyRAM still has better performances than ARM when the latter can use more than 50% of cache.

5 Conclusions

We presented the DyRAM framework for handling efficiently multicast communications on computational grids. The architecture consists of a core protocol and active services located at the edge of the backbone network. These services are the packet losses detection, the NACK suppression, the subcast and the replier election. Simulation results show that local recovery, associated with early loss detection can provide both low-latency and increased bandwidth compared to a traditional solution.

The network model we considered assumed that there is a single receiver attached to a downstream link of an edge active router. If there are multiple receivers attached through a non active router then more than one receiver may receive a NACK packet (because the replier election is performed on a link basis) and believe to be the elected replier, so duplicate repairs may be sent. There are no simple ways to suppress the duplicate repairs sent by these receivers before they reach the active router. In this kind of topology, we suggest to put active routers as near as possible to the receivers, especially when there are a significant number of receivers. An example of such a situation is the case of several computing centers, each of them with an access router and several attached receivers, located on the same campus in a grid-like environment. In this case, it is preferable to have an active router for each computing center. For a few number of receivers, this problem is not very harmful.

One of our main concerns while designing DyRAM was to propose low-overhead (in terms of processing and memory requirements) and easy to implement solutions for the active services to be put in routers. Based on small data structures we believe the active services introduce only low overheads in routers and that implementations without much performance degradation on the routing and forwarding functions are possible.

References

1. M. Litzkow and M. Livny. Experience With The Condor Distributed Batch System. In *IEEE Workshop on Experimental Distributed Systems, October 1990.*
2. I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *Intl J. Supercomputing Applications, 11(2):115-128, 1997*
3. A. Grimshaw, A. Ferrari, F. Knabe and M. Humphrey. Legion: An Operating System for Wide-area computing. *IEEE Computer, 32(5):29-37, May 1999*
4. XTP Forum. *Xpress Transport Protocol Specification*, March 1995.
5. S. Floyd, V. Jacobson, and Liu C. G. A reliable multicast framework for light weight session and application level framing. In *ACM SIGCOMM'95*, pp342–356.
6. S. Paul and K. K. Sabnani. Reliable multicast transport protocol (rmtp). *IEEE JSAC, Special Issue on Network Support for Multipoint Comm.*, 15(3):407–421.
7. R. Yavatkar, J. Griffioen, and M. Sudan. A reliable dissemination protocol for interactive collaborative applications. In *ACM Multimedia'95.*
8. Christos Papadopoulos, Guru M. Parulkar, and George Varghese. An error control scheme for large-scale multicast applications. In *IEEE INFOCOM'98*, pp1188–1996.
9. T. Speakman et al. Pgm reliable transport protocol specification. internet draft.
10. D. L. Tennehouse et al. A survey of active network research. *IEEE Comm. Mag.*, pp80–86, January 1997.
11. L. Wei, H. Lehman, S. J. Garland, and D. L. Tennenhouse. Active reliable multicast. In *IEEE INFOCOM'98.*
12. S. K. Kasera et al. Scalable fair reliable multicast using active services. *IEEE Networks, Special Issue on Multicast*, 2000.
13. S. Zabele et al. Improving Distributed Simulation Performance Using Active Networks. In *World Multi Conference, 2000.*
14. M. Maimour and C. Pham. A Throughput Analysis of Reliable Multicast Protocols in an Active Networking Environment. In *6th IEEE Symp. on Comp. and Comm., 2001.*
15. M. Maimour and C. Pham. An analysis of a router-based loss detection service for active reliable multicast protocols. Technical report, RESAM, http://www.ens-lyon.fr/~mmaimour/Paper/TR/TR03-2001.ps.gz, November 2001.
16. R. Bagrodia et al. Parsec: A parallel simulation environment for complex systems. *Computer Magazine, 31(10), October 1998, pp77-85.*
17. C. Papadopoulos and E. Laliotis. Incremental Deployment of a Router-assisted Reliable Multicast Scheme. *Proc. of NGC 2000 Workshop*