A Loss Detection Service for Active Reliable Multicast Protocols

M. Maimour and C. D. Pham

RESAM, Lyon1. ENS, 46 allee d'Italie
69364 Lyon Cedex 07 - France
email:{mmaimour,cpham}@ens-lyon.fr

Abstract

Reliable multicast protocols have gained popularity with active service contributions where routers implement additional functionalities. Reducing the delay of recovery is one of the desirable features of a reliable multicast protocol. In this paper we propose an active-based architecture with specialized routers. Using simulations we show how this architecture with the proposed services (mainly the loss recovery from the receivers and the loss detection at the routers), could improve the performances of a reliable multicast application in term of the recovery delay.

Keywords

Reliable Multicast, Active Networks.

1 Introduction

Multicast handles one-to-many communications with one sender transmitting the same data to more than one receiver. At the network level IP multicast provides an efficient one-to-many IP packets delivery but without any reliability guarantees. However data dissemination applications such as distributed computing or interactive simulations require a reliable transfer. In early ACK-based protocols, the sender is responsible for both the loss detection and the recovery (Strayer et al., 1992). These protocols do not scale well to a large number of receivers due to the ACK implosion problem. The use of NACKs instead of ACKs moves the loss detection responsibility to the receivers. The problem then turns into a NACK implosion problem when a large number of receivers have subscribed to the multicast session. The next step in improving scalability is the use of the local recoveries where the retransmission of a lost packet can be performed by some other nodes in the multicast tree (Floyd et al., 1997; Paul and Sabnani, 1997; Yavatkar et al., 1995; Papadopoulos et al., 1998; Speakman et al., 1998; Holbrook et al., 1995). Local recoveries can dramatically decrease the recovery latency. There are basically 2 possibilities for enabling local recoveries: (i) use some receivers or dedicated servers as repliers, or (ii) use network elements such as routers. For instance, the replier could be any receiver in the neighborhood (SRM (Floyd et al., 1997)), a designated receiver (RMTP (Paul and Sabnani, 1997), TMTP (Yavatkar et al., 1995), LMS (Papadopoulos et al., 1998), PGM (Speakman et al., 1998)) or a logging server (LBRM (Holbrook et al., 1995)) in a hierarchical structure.

Active networks (Tennehouse et al., 1997) open new perspectives in providing more efficient solutions for the problem of the feedback implosion. Active approaches benefit from the assistance of the routers which are able to perform customized computations on the messages flowing through them. ARM (Active Reliable Multicast) (Lehman et al., 1998) is one example of such approaches that was recently proposed in the research community. The main contribution of active services is a *best-effort* cache of data packets to allow the active router to retransmit the lost data packet instead of the source. 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 practice, most of router's caching means are limited and the routers must support many sessions in parallel. Using a replier has the main advantage of requiring a memory usage as low as possible within network elements and is potentially more scalable. DyRAM (Dynamic Replier Active Reliable Multicast) (Maimour and Pham, 2002) is an other active approach that provides a solution to the problem of scalability at the routers by enabling local recovery from the receivers without any caching at the routers.

In (Maimour and Pham, 2001) we proposed and analyzed a new active service that consists in the loss detection by the routers themselves. In this case, routers are capable to detect packet losses and consequently generate corresponding NACKs to be sent to the source. In this paper, some results of this analytical study will be exploited to propose an active-based reliable multicast architecture more dedicated to our DyRAM protocol. The rest of the paper is organized as follows. Section 2 gives an overview of the DyRAM protocol. Section 3 presents the loss detection service. In section 4, our active-based architecture is presented. Section 5 shows how the loss detection service combined with the proposed architecture could improve the DyRAM protocol. Section 6 concludes the paper.

2 DyRAM: an overview

In this section, an overview of DyRAM is given. For a more detailed description, the reader is referred to (Maimour and Pham, 2002). DyRAM is based on a tree structure constructed on a per-packet basis with the assistance of the routers. The protocol uses a NACK-based scheme with a receiver-based local recovery where receivers are responsible for both the loss detection and in some cases the retransmission of repair packets. A receiver detects a loss by sequence gaps and upon the detection of a loss, 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. In order to limit the processing overheads of duplicate NACKs and to avoid the corresponding retransmissions, the source, the active routers and the receivers ignore similar NACKs for a certain period of time. Routers maintain information about NACKs flowing through them. For each received NACK, the router creates or simply updates an existing NACK state (NS) structure. Such a structure contains during its life time the following information:

- seq: the sequence number of the requested packet,
- $time$: the time when the last valid NACK for this packet has been received and forwarded toward the source,

- $rank$: 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.

2.1 NACK suppression and subcast

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 (not a duplicate one), and if so, the router will forward it on the elected replier link. Otherwise this NACK will serve to properly update the NS structure (rank and subcast list) and is dropped afterward.

The subcast list in the NS structure 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.

2.2 Replier election

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.

In order to decrease the overhead (longer recovery latency) introduced by the DTD timer 2 optimizations are proposed. First, when a router receives NACK packets from all of the downstream links before the expiration of the DTD timer it will immediately forward the last NACK received toward the source and will cancel the replier election process. The second optimization consists in keeping track within a router of the received data packets in order to quickly detect packet losses occurring from upstream and affecting all its subtree. This can be done simply by maintaining a track list structure (TL) for each multicast session handled by the router. A TL has three components:

- $lastOrdered$ is the sequence number of the last data packet received in order. All packets with a sequence number less or equal to $lastOrdered$ have definitely been received by the router.

- $lastReceived$ is the sequence number of the last received data packet. All packets with a sequence number greater than $lastReceived$ have not been received by the router.
- $lostList$ contains the list of data packets not received by the router with sequence number greater than $lastOrdered$ and less than $lastReceived$. This list is empty when ($lastReceived < lastOrdered + 1$) and contains at least one element otherwise.

A router maintaining such a track list (TL) structure is able to decide in some cases to forward the NACK immediately toward the source instead of waiting for the expiration of the DTD timer. These cases include the case when the requested data packet sequence number is greater than $lastOrdered$ and contained in the $lostList$. The subcast list is updated so that the repair packet (from the source) would be forwarded on all downstream links.

3 The active loss detection service

Earlier reliable multicast protocols put the burden of both the loss detection and the recovery at the sender side. These protocols are TCP-like and use ACKs combined with timers to detect losses. These ACK-based schemes violate the IP-Multicast Model (Deering and Cheriton, 1990) where the source is not aware of the identity of the receivers. The use of NACKs instead of ACKs moves the loss detection responsibility to the receivers. A receiver detects a loss on receipt of an out-of-order data packet or on a timeout expiration. When a loss is detected, the receiver would send a NACK to the source requesting the lost data packet.

The repair latency can be reduced if the lost packet could be requested as soon as possible. This can be achieved by enabling routers to detect losses and therefore to generate NACKs to be sent to the source. An active router detects a loss when a gap occurs in the data packet sequence. On a loss detection, the router would generate immediately 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 period, all NACK packets received for this data packet from the downstream links are ignored. The loss detection service can be implemented without adding any more soft state at the routers. The NS and the TL structures are sufficient to implement this service. With respect to the main algorithms of DyRAM, only the data packet service is modified to take into account the loss detection service. When an active router receives a data packet, it will execute the data packet service given below :

```
Update the $TL$ $trackList$ of the session
if DP is a repair then
  Look for the corresponding NACK state structure $NS$
  if such a structure exists then
    send DP through links in $NS.subList$
    free $NS$
  else if the DP arrived from the upstream link then
    forward DP on all the downstream links
  else {the DP arrived from downstream, link $i$}
    forward the DP on all the downstream links except link $i$
  end if
```

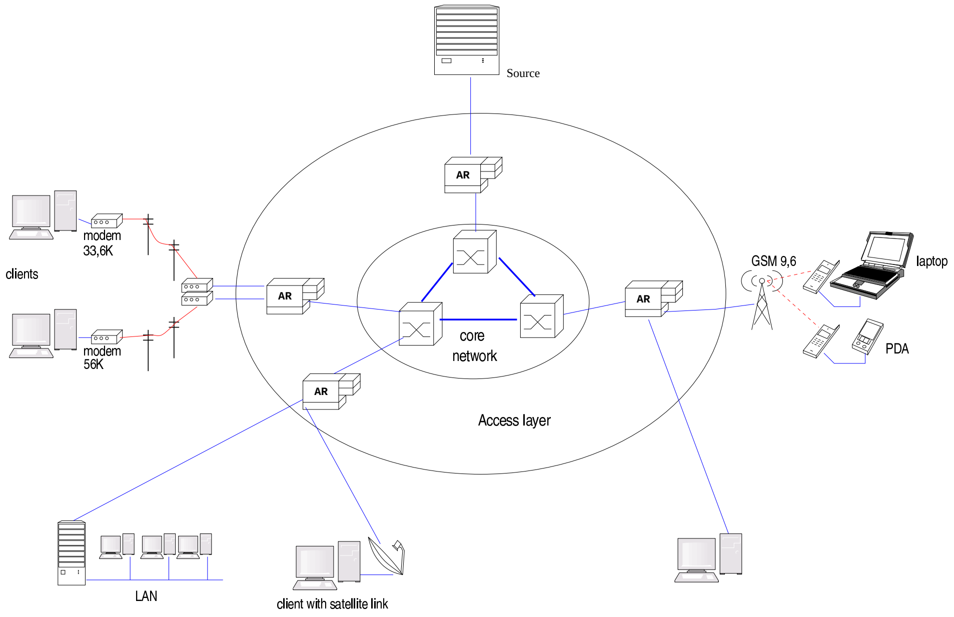


Figure 1: Active-based reliable multicast architecture

```
else
  Forward DP on all the downstream links
  for each $DP_{lost}$ with $((DP_{lost}.seq > TL.lastOrdered + 1)$ and $(DP_{lost}.seq \in TL.lostList))$ do
    send a NACK($DP_{lost}.seq$) to the source
    Create or update the corresponding $NS$ structure $NS_{lost}$:
    $NS_{lost}.seq \leftarrow DP_{lost}.seq$
    $NS_{lost}.time \leftarrow current\ time$
    $NS_{lost}.subList \leftarrow$ all the links downstream
  end for
end if
```

4 An active-based reliable multicast architecture

We consider the IP multicast model introduced by Deering (Deering and Cheriton, 1990). The source sends messages to a multicast address subscribed to by all the receivers. The sender need not know the receivers identities. A receiver can join or leave a multicast session as it wishes by simply sending a *join* or a *leave* message via IGMP (Internet Group Management Protocol). The routers has to make their best effort to deliver the traffic from the sender to each receiver. The packets will be duplicated in the network as needed thus reducing bandwidth consumption compared to the unicast approach. Our reliable protocol DyRAM works on top of UDP and will guarantee that all the sent packets are received by all the receivers of the multicast group.

Our architecture (see figure 1) is based on a virtual topology of active routers which are able to perform customized computations (services) on the messages (data packets and NACKs) flowing through them. We only consider active routers at the edge of the core network. This is due to the fact that the core network is reliable and a very high-speed network. Adding complex processing functions inside the core network will certainly slow down the packet forwarding functions.

For the realization of our active approach, we propose the *programmable switch approach* which maintains the existing packet format. 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. Consequently a data packet would be processed by a data packet service and a NACK would be processed by a NACK service.

To dynamically handle the multicast group topology changes, active routers have to be able to add or remove the required 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 easily download the required services. This can be achieved by requesting the services from the closest active router which has already installed them. If there is no such active router then the services need to be sent from the source. The required services are always transmitted via a reliable channel. These operations are usually handled by the active execution environment.

All the active routers are assumed to perform at least two active services, the NACK suppression service 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*. As have been shown in (Maimour and Pham, 2001), 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 should only perform the replier election service as seen in the previous section.

5 Simulation results

A set of simulations are performed to show how a loss detection service could decrease the delay of recovery of an active reliable multicast protocol. To do so, four protocols noted A, D, $DyRAM$ and $DyRAM^+$ are simulated on a network model derived from the proposed architecture. In addition to the source active router A_S, we consider N active routers A_i, $i \in \{1 \dots N\}$. Each active router A_i is responsible of B receivers forming a local group. All of the four protocols benefit from the NACK suppression and the subcast services. Whereas A only benefits from these two services, D benefits from the loss detection service at the source router. $DyRAM$ is similar to DyRAM where local recoveries from the receivers are possible. $DyRAM^+$ behaves like $DyRAM$ except that additionally the source router performs the loss detection service. In our loss model, we consider both the spatial and the temporal correlation of data packet losses. The spatial correlation is introduced by considering a per-link loss rate and the core network is considered reliable. The temporal correlation of losses is achieved by using the same model as in (Maimour and Pham, 2002). We also consider that there is l_b backbone links between the source router A_S and every active router A_i. The simulations are implemented using the PARSEC language developed at UCLA (Bagrodia et al., 1998).

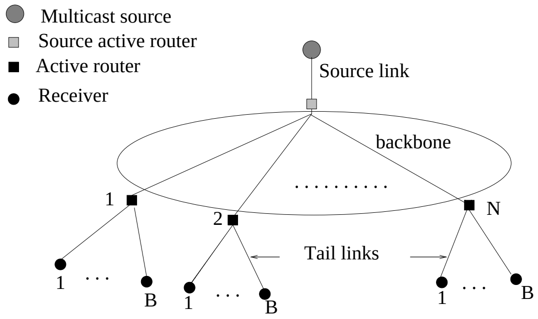
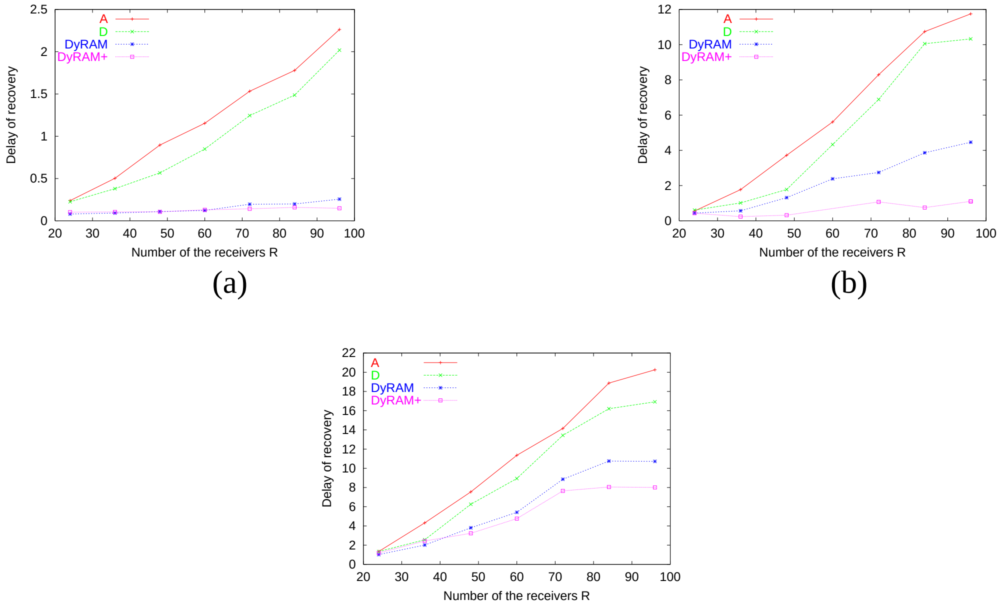


Figure 2: Network model

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. 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. These values are derived from measures in (Lehman et al., 1998).



(a)

(b)

(c)

Figure 3: The recovery delay with (a) $p = 0.05$**, (b)** $p = 0.25$ **and (c)** $p = 0.50$

Figure 3 plots the recovery delay (normalized to the RTT) for the four protocols as a function of the number of the receivers for different loss rates. First of all, it is noticeable that protocols $DyRAM$ and $DyRAM^+$ with local recovery from the receivers always perform better. For instance, we can see in figure 3(a) that $DyRAM$ goes up to 10 times faster than A for a loss rate of 5%. Now, when the loss detection service is applied to A (giving protocol D) the recovery delay can be reduced. In fact as we can see for the different loss rates, D always performs better than A thanks to the loss detection service. When applying the loss detection service to $DyRAM$, the delay of recovery decreased mainly for high loss rates and a large number of receivers. For instance, the loss detection service allows DyRAM to go 4 times faster for 96 receivers and a loss rate of 25%. We can also notice in figures 3(a)(c) that $DyRAM$ slightly performs better than $DyRAM^+$ when the number of receivers is small. Therefore it is unjustified to perform the loss detection service for a few number of receivers since the local recovery is sufficient to reduce the recovery delay. This does not appear to be a limitation of the loss detection service since a multicast session has generally to support a large number of receivers.

6 Conclusions

In this paper, we proposed an active-based reliable multicast architecture with specialized routers more dedicated to our DyRAM protocol. Adding a loss detection service to the DyRAM protocol at the source router helps to reduce the delay of recovery without overwhelming the other active routers that perform the replier election service. Simulation results have shown that the DyRAM protocol performs better with the loss detection service especially for big loss rate when increasing the number of the receivers.

References

Bagrodia, R. et al. (1998). Parsec: A parallel simulation environment for complex systems. *Computer Magazine*.

Deering, S. E. and Cheriton, D. R. (1990). Multicast routing in datagrams internetworks and extended lans. In *ACM Transactions on Computer Systems*, pages 85–110.

Floyd, S., Jacobson, V., Liu, C.-G., McCanne, S., and Zhang, L. (1997). A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5(6):784–803.

Holbrook, H. W., Singhal, S. K., and Cheriton, D. R. (1995). Log-based receiver-reliable multicast for distributed interactive simulation. In *SIGCOMM*, pages 328–341.

Lehman, L., Garland, S., and Tennehouse, D. (1998). Active reliable multicast. In *Proceeding of the IEEE INFOCOM, San Francisco, CA*.

Maimour, M. and Pham, C. (2002). Dynamic replier active reliable multicast (dyram). In *To appear in Proceedings of the Seventh IEEE Symposium on Computers and Communications (ISCC 2002)*.

Maimour, M. and Pham, C. D. (2001). 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.

Papadopoulos, C., Parulkar, G. M., and Varghese, G. (1998). An error control scheme for large-scale multicast applications. In *Proceeding of the IEEE INFOCOM*, pages 1188–1996.

Paul, S. and Sabnani, K. (1997). Reliable multicast transport protocol (RMTP). *IEEE journal of Selected Areas in Communication, Special Issue on Network Support for Multipoint Communications, Special ISSue on Network support for Multipoint Communication*, 15(3):407–421.

Speakman, T. et al. (1998). Pgm reliable transport protocol specification. internet draft.

Strayer, T. W., Dempsey, B. J., and Weaver, A. C. (1992). XTP – THE XPRESS TRANSFER PROTOCOL. *Addison-Wesley Publishing Company*, page 580 pages.

Tennehouse, D. L., Smith, J. M., Sincoskie, W. D., Wetherall, D. J., and Winden, G. J. (1997). A survey of active network research. *IEEE Communication Magazine*, pages 80–86.

Yavatkar, R., Griffoen, J., and Sudan, M. (1995). A reliable dissemination protocol for interactive collaborative applications. In *ACM Multimedia*, pages 333–344.