Light-Weight Randomized Reliable Multicasting Protocol

Nipoon Malhotra, Shrish Ranjan, Saurabh Bagchi
Dependable Computing Systems Laboratory, Purdue University
E mail: {nmalhot, sranjan, sbagchi}@purdue.edu

1. Introduction

Multicasting is an efficient way of distributing data from a sender to multiple receivers. There has been considerable interest in augmenting the best-effort nature of IP multicast protocols to support reliable multicast capable of tolerating node crashes and message losses. The basic tree-based protocols (RMTP [4], TRAM [5]) suffer from the problem that under failures, a local designated recovery host may get overloaded, and costly remote recovery may be performed even if a host in the local region has the message being requested. A protocol proposed to solve the problems is the Randomized Reliable Multicast Protocol (RRMP) [1]. Buffer management techniques proposed for RRMP [2] involve a trade off between lost message recovery latency and the amount of the buffer space used. In this paper, we propose a protocol called Light-weight Randomized Reliable Multicast (*LRRM*) which uses an alternative lost message recovery strategy that leads to lesser buffer requirement without compromising the recovery latency. As the reliable multicast protocols are deployed over wide area networks, it is a likely scenario that the intermediate nodes are light-weight and constrained in their buffer space and processing capabilities. Also the receivers may have widely varying reception rates and periods of disconnection resulting in large buffer space requirements. This motivates LRRM.

2. RRMP protocol

In RRMP, receivers are divided into a number of regions based on their distance from the sender. Receivers maintain group membership information of their own as well as parent region by exchange of session messages. Two types of buffering schemes are used, short term and long term. All received messages are stored in the short term buffer for a certain time *T*. If no retransmission requests are received for a message within a time interval *T*, a decision is made to either discard this *idle message* or store it in the long term buffer with a probability *P*. Eventually, a message for which no retransmission requests have been received is deleted. The expected numbers of buffers used in a region is *C.*

The onus of recovery of lost messages lies with the receiver. On detecting message loss, a receiver concurrently initiates the local and the remote recovery procedure. In local recovery, a receiver *p*, randomly chooses another member *q* of its region and sends a request for transmission of the lost message. Simultaneously, a timer is started. If *q* has buffered the message, it responds by sending a unicast reply. Otherwise, *p* will time out and choose another member for a retransmission request. In the remote recovery procedure, *p* chooses a member *r* at random from its parent region and requests the lost message.

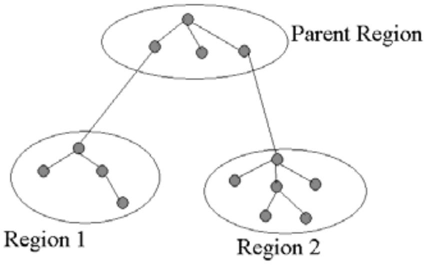

Fig 1: Hierarchical regions in RRMP

3. Proposed recovery strategy

In RRMP, a receiver searches a lost message using a series of unicast requests. To achieve reasonable recovery latency, a sizeable number of members of a region must store messages in long term buffers.

The proposed strategy involves sending a multicast request for a lost packet to all members of the local region. Local members, who have the message, respond to this request by unicast replies. The lost message can be recovered in a reasonable time even if a single copy of the same is present in the local region

If p_i is the probability of finding a message in the buffer of a member *i* of group and T_i is the upper bound on the time required to recover a message from the member *i*, the expected value of recovery latency is given by

$$p_1T_1 + p_2(1-p_1)(T_1 + T_2)+ \dots + p_N(1-p_1)\dots(1-p_{N-1})\,(T_1+ \dots + T_N)$$

where *N* is the number of members in a certain region. p_i is dependent on *P* and the message loss rate *l.* Assuming $p_i = p$ and the upper bound on recovery time from the i^{th} member polled be $T_i=T_0$ then the expression for average latency of local recovery is given by

$$T_0/p^* \{(1-(1-p)^{N+1}) - (N+1)\, p\, (1-p)^{N}\}$$

For a multicast based local recovery the upper bound is simply T_0. This corresponds to the best case local recovery latency for RRMP. Local recovery will be successful even if a single member has buffered a copy of the lost message. Therefore, we can considerably reduce the probability *P* of buffering messages in long term buffers. This translates to a reduced buffer requirement *C*.

However, using multicast requests for error recovery can lead to message flooding. If multiple members of the region loose the same packet then they would flood the network with redundant multicast requests. To reduce this problem a back- off algorithm is used. When a node detects a message loss, it waits for a random time before multicasting a request for the message. Other members who hear a multicast for a message that they themselves have not received suppress their own multicast. Instead, they send a unicast request to the originator of the multicast request for the lost packet (Fig 2a). Now it becomes the responsibility of the originator of the multicast request to deliver this packet to other requesting nodes once it finds it (Fig 2b).

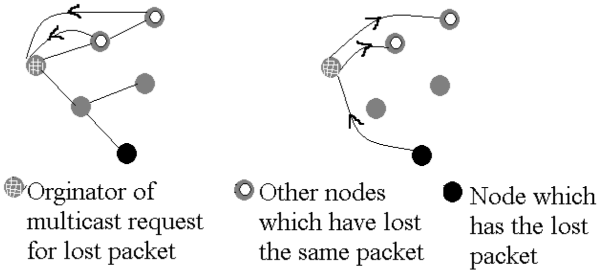
Orginator of multicast request for lost packet
Other nodes which have lost the same packet
Node which has the lost packet

Fig 2a, 2b: Local Recovery Process

Because of the nature of the proposed solution, even a single copy of a lost message in a region is sufficient to satisfy a recovery request in a reasonable time. This time is given by $T0 + \Delta T$ where ΔT is the average time for random back off. This parameter is dependent on the probability distribution used to implement back off.

We intend to use a hash function to ensure with a high probability that at least one node in each region has the message. Nodes will use message identifiers and their own id to decide whether they are eligible to buffer a certain message. Based on the result, they can decide to store a message with a probability *P*. The proposed solution does not try to store copies of all the packets in a region because with moderate packet loss rates, retransmission requests for most of the packets will never be issued. The value of probability *P* is a configurable parameter chosen based on packet loss rates. The concept of short term and long term buffer from RRMP has been replaced by a single buffer.

We have assumed a dynamic region membership because of which the unique node which would have stored the packet cannot be identified without storing the entire membership history. Thus unicasts can't be used to request lost packets as in [3]

If a packet is not found among the members of a local region then a unicast request for it is sent to a randomly chosen member of the parent region. This parent follows the same multicast based recovery algorithm in its region. This procedure can continue in a recursive fashion and in the worst case, the message can be retrieved from the original sender.

4. Experiments

We have developed a simulation model for LRRM in NS-2. We plan to simulate LRRM and obtain empirical results relating latency to buffer requirements. We will compare the results on buffer utilization, latency, andmessage traffic with those from RRMP.

5. References

[1] Zhen Xiao and K. Birman, "A Randomized Error Recovery Algorithm for Reliable Multicast", IEEE Infocom April 2001, Alaska.

[2] Xiao Zhen, K.P. Birman, R. van Renesse, "Optimizing buffer management for reliable multicast" Proceedings of the International Conference on Dependable Systems and Networks (DSN '02), June 2002.

[3] Ozkasap, Oznur, van Renesse, Robbert, Birman, Kenneth and Xiao, Zhen, "Efficient Buffering in Reliable Multicast Protocols", Proceedings of the First Workshop on Networked Group Communication. (NGC99) Pisa, Italy. (November 1999).

[4] Sanjoy Paul and John C. Lin, "RMTP: A Reliable Multicast Transport Protocol", INFOCOMM 1996, pp.1414-1424.

[5] Dah Ming Chiu, Stephen Hurst, Miriam Kadansky and Joseph Wesley, "TRAM: A Tree-based Reliable Multicast Protocol", Sun Technical Report TR 98-66, July 1998.