

# ON LARGE-SCALE RELIABLE MULTICAST PROTOCOLS

M Schuba, P Reichl

Aachen University of Technology, Germany

In this paper we examine the impact of widely spread multicast groups on the performance of reliable multicast protocols. In a brief analysis we show that transmission cost for reliable multicast increases rapidly with the distance between retransmitting and receiving hosts. Since this distance is not considered by existing protocols, which are based on retransmissions originated at the sender or receiver, the performance often is rather poor and a new error recovery concept has to be developed. Our approach, the Scalable and Reliable Multicast Transport protocol (SRMT), allows retransmissions by extended multicast routers to limit the path length for error recovery. Moreover, the retransmission load can be further reduced by an XOR combination of packets. SRMT is defined as an overlay network model and therefore can be applied to any existing multicast protocol.

## 1 INTRODUCTION

The availability and interconnection of communication networks worldwide has led to a global information exchange with growing demand for efficient network services and applications. In the last years developments in multicast communications, e.g. IP multicast (see Deering (1), Eriksson (2), or Hermanns and Schuba (3)), have made groupware applications like videoconferencing very popular. The multicast service offered by today's networks is usually best effort, i.e. no guarantees concerning loss, duplicates etc. are given. Since a number of applications cannot accept any packet losses, reliable multicast protocols have been implemented for error recovery (see e.g. Armstrong et al (4) or Whetten et al (5)). Unfortunately most of these concepts scale bad, i.e. they perform very poor if the number of receivers grows or if members are far away from the sender. However, multicast groups of some applications, e.g. electronic newspapers, distributed simulation, software distribution, or (of topical interest) WWW push operations, may consist of thousands of members distributed all over the world. Therefore usual concepts for error recovery cannot be applied any more. Only recently some protocols for large-scale reliable multicast have been proposed (see Heinrichs (6), Floyd et al (7), Hofmann (8), Paul et al (9), Yavatkar et al (10), or Kasera et al (11)). The techniques used by these protocols include reduction of acknowledgement (ACK) implosion and local or restricted retransmissions.

In this paper we demonstrate that the efficiency of existing concepts for large-scale reliable multicast is often still rather poor. A simple analysis illustrates that the number of required transmissions for large and widely spread groups becomes unacceptable even for small packet loss probabilities. A major improvement can only be achieved by a reduction of the path lengths between sending and receiving nodes. Therefore we propose a new large-scale reliable multicast protocol (SRMT, Scalable and Reliable Multicast Transport), which allows retransmissions of lost packets by routers and thus limits the length of retransmission paths. In addition a distributed bucket algorithm in combination with XOR-based retransmissions reduces the multicast traffic. Since our solution operates as an overlay network it can be applied to any kind of packet switched or virtual-circuit network.

The paper is structured as follows. After a brief classification of existing protocols for large-scale multicast (chapter 2) we analyze the number of transmissions required for so-called sender-originated reliable multicast with respect to link loss probability, size of the multicast group and distance to the group (chapter 3). In chapter 4 we apply our analysis to scenarios of the protocol classes previously defined. Based on these results we define the SRMT protocol as new approach for reliable multicast in wide area networks (chapter 5). In chapter 6 the main results are summarized and a short outlook on future work is given.

## 2 CLASSIFICATION OF LARGE-SCALE RELIABLE MULTICAST PROTOCOLS

Several protocols have been proposed for scalable, reliable multicast in the last years, e.g. AMTP (6), SRM (7), LGC (8), RMTP (9), TMTP (10), or a protocol based on multiple multicast groups (11). For our purposes we classify the protocols with regard to their retransmission strategies.

### Class 1: Sender-Originated Retransmissions

Protocols of this class try to improve scalability by usual ACK reduction techniques, e.g. NAKs, aggregated ACKs or NAKs, or group ACKs. Retransmissions are multicasted by the sender to the whole group. A typical example for a protocol of Class 1 is AMTP.

## Class 2: Sender-Originated Retransmissions Using Multiple Multicast Groups

Again all packets are retransmitted by the sender. In contrast to Class 1 protocols a separate multicast group is used for every lost packet. A host missing a packet signals this by a NAK and joins the respective “retransmission channel”. Thus processing cost at the receiver and network load can be decreased. Three Class 2 protocols are proposed in (11).

## Class 3: Dedicated Receiver-Originated Retransmissions

Here receivers of the same region form a local group. A dedicated receiver (DR) of each group is responsible for processing of ACKs and starting of retransmissions if one of the local group members detects a packet loss. The groups can be structured hierarchically in order to make the approach more scalable. In this case several levels of DRs exist. Representatives of Class 3 protocols are LGC, RMTP, and TMTP.

## Class 4: All Receiver-Originated Retransmissions

Class 4 protocols allow any member of the group to react on retransmission requests. Therefore these requests have to be sent to the multicast group. A host that is able to recover a packet loss retransmits the packet to the group. The number of duplicate requests and retransmissions can be restricted by using a limited scope for all messages and randomly delaying transmissions. SRM is based on this retransmission strategy.

## 3 TRANSMISSION COST ANALYSIS

Large-scale multicast differs from local multicast in the following features:

- the number of receivers can be very large, i.e. a multicast group may consist of several thousand members,
- group members can be distributed worldwide, i.e. some or all receivers might be far away from the source,
- the quality of connections might be very bad due to some overloaded or unreliable links.

For an analysis of the transmission cost these characteristics can be expressed by three parameters:

$D$  = number of destinations,

$h$  = number of hops between source and receivers,

$p$  = cell loss probability per hop.

We assume  $h$  to be constant and  $p$  to be equal for all links.

Let  $p_B$  denote the probability that a packet is lost on a branch in the multicast tree from the sender to a receiver. Then  $p_B$  can be expressed as

$$p_B = 1 - (1 - p)^h$$

Let the random variable  $X$  denote the number of transmissions by the sender necessary for all  $D$  receivers to successfully receive a packet. Assuming all retransmissions to be performed by the sender and loss events independent for each receiver the following probabilities for  $X$  can be derived (cf. Pingali et al (12)):

$$P(X \leq n) = (1 - p_B^n)^D$$

and thus

$$P(X = n) = (1 - p_B^n)^D - (1 - p_B^{n-1})^D. \quad [1]$$

Fig. 1 - 3 show how this density function varies in dependence of  $h$ ,  $p$  and  $D$ . Note that although  $X$  is a discrete random variable we use continuous plots in order to distinct clearly between different density functions.

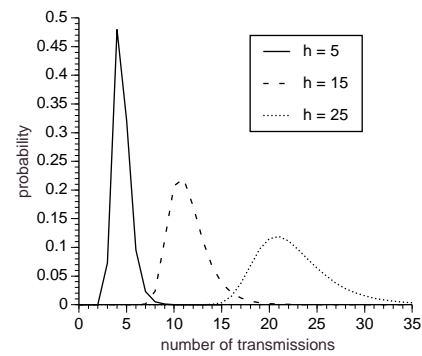


Figure 1. Density of the number of transmissions ( $D = 1000$ ,  $p = 0.05$ )

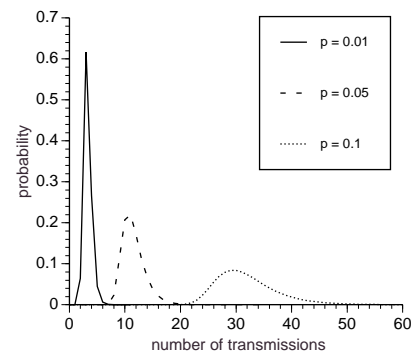


Figure 2. Density of the number of transmissions ( $D = 1000$ ,  $h = 15$ )

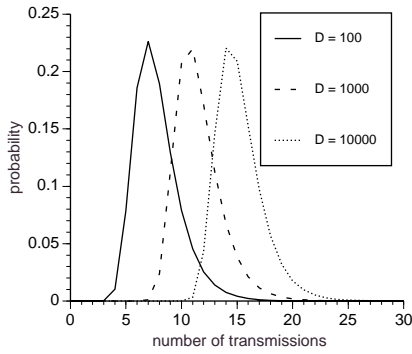


Figure 3. Density of the number of transmissions ( $h = 15, p = 0.05$ )

The number of transmissions grows rapidly with an increasing number of hops (fig. 1) or with increasing loss probability (fig. 2). However, the influence of parameter  $D$  is not as significant as might be expected (fig. 3). Even if the group size grows to 10,000 the increase of the number of required transmissions is marginal compared to smaller groups. Numerically calculated expectations confirm these conclusions. Fig. 4 and 5 show the expected number of retransmissions relative to a realistic link loss probability (up to 10%).

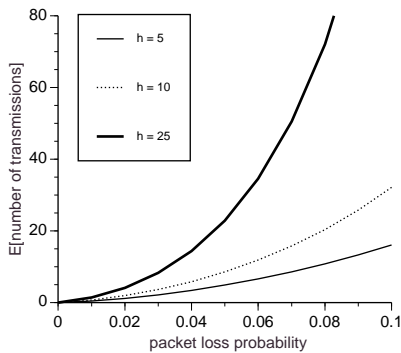


Figure 4. Expected number of transmissions ( $D = 10$ )

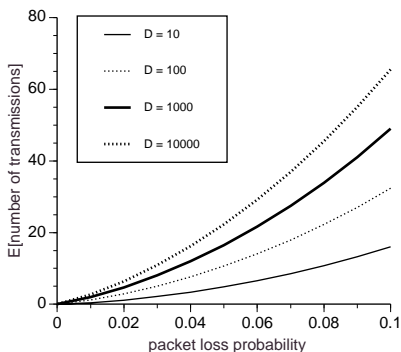


Figure 5. Expected number of transmissions ( $h = 5$ )

Incrementing the hop distance between sender and receivers results in a significant rise in the expected number of transmissions (even for small groups of  $D =$

10 receivers). In comparison to this the expectation grows marginally even if the number of receivers is increased by factor 10.

#### 4 TRANSMISSION COST OF EXISTING PROTOCOL CLASSES

Obviously the assumptions made for our analysis (all receivers have the same distance to the sender and all retransmissions are sender-originated) do not hold for all protocol classes and possible multicast groups. Nevertheless, equation [1] can be used to approximate the transmission cost for the different protocol classes. For each class we can construct a worst case scenario where our analysis can be applied.

Class 1 and 2 both use retransmissions from the sender to recover from packet losses. In the worst case many receivers have a large distance to the sender (see fig. 6).

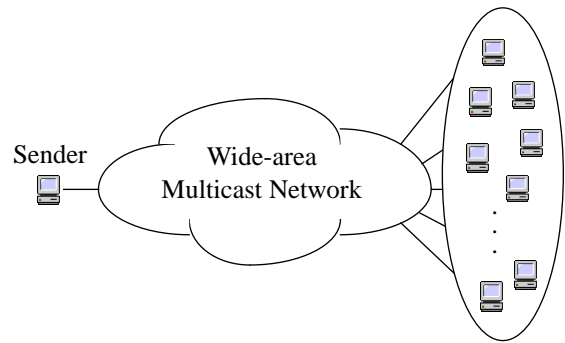


Figure 6. Worst case for class 1 and class 2 protocols

The expected number of transmissions (calculated based on equation [1] with  $h$  and  $D$  very large) in such a scenario would be worse than the curves shown in fig. 4 and 5. Thus class 1 and 2 protocols are unacceptable even for networks with small loss probabilities.

In Class 3 retransmissions are restricted to a (hierarchical) local group. A worst case scenario can be easily constructed. Consider a large number of first level DRs very far away from the sender (see fig. 7). This might be a transmission from the U.S. (at night) to a large group in the remaining world (at work).

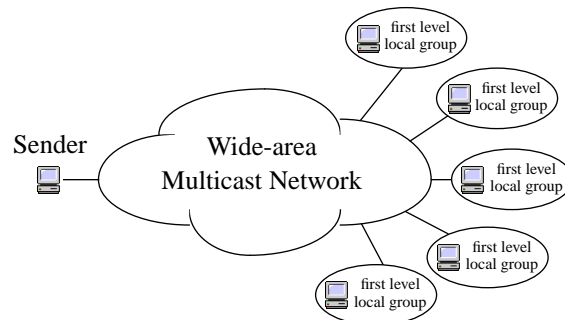


Figure 7. Worst case for class 3 protocols

The expected number of transmissions by the sender can be calculated using equation [1] with parameter  $D$  set to the number of first level DRs and  $h$  taking a large value. The expectations in fig. 4 ( $D = 10$  and  $h = 25$ ) show that for such (quite realistic) scenarios the performance of class 3 protocols is rather poor with respect to the transmission cost.

In a Class 4 protocol each receiver is responsible for retransmissions. The worst case for this protocol is a (sub)group of members with wide geographic spread (see fig. 8).

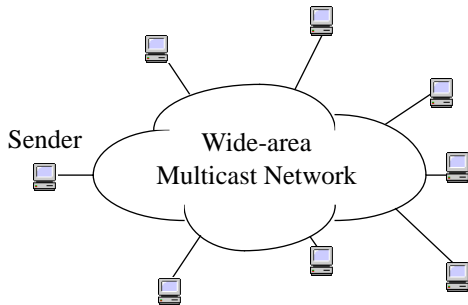


Figure 8. Worst case for class 4 protocols

We assume the distance between any two receivers to be larger than their distance to the sender. Thus each retransmission is performed by the sender itself and equation [1] can be applied just by adapting parameter  $h$ , resp. Since the value of  $h$  decreases with a growing number of group members even in this worst case, it will probably not be too large. We think that for a group of size 10 values of  $h$  between 10 and 25 (cf. again fig 4) are realistic. Although this yields a slightly better performance of class 4 protocols in comparison to class 3 protocols the results are still unsatisfying. Moreover, class 4 protocols are based on complex timer mechanisms to handle retransmissions. These mechanisms might not work well in widely spread groups and may therefore cause additional overhead, e.g. duplicate retransmissions.

## 5 SRMT - SCALABLE AND RELIABLE MULTICAST TRANSPORT

The previous chapter has shown that high transmission cost of existing reliable multicast protocols is mainly caused by large distances between hosts retransmitting a packet and their receivers (large  $h$  values). Because the location of multicast group members cannot be influenced by the protocol a reduction of transmission cost can only be achieved by a protocol modification. We propose the SRMT protocol for large-scale reliable multicast, which uses router-originated retransmissions and thus allows packets to be retransmitted exactly on the path of the multicast tree where they were lost. This minimizes not only the value of parameter  $h$  but additionally the number of receivers  $D$  of a retransmitting host. Based on the results of chapter 3 and 4 we expect that transmis-

sion cost for SRMT will be very low in comparison to the cost of existing protocols.

We define SRMT as an overlay network on top of an existing best effort multicast protocol. This allows routers to be upgraded step by step. All SRMT nodes involved in a multicast transmission for a group (i.e. sender, routers and receivers) build a logical multicast tree rooted at the sender. Each SRMT node, which has a child in the multicast tree, acts as an SRMT sender, each SRMT node with a parent node as an SRMT receiver. An SRMT sender together with all its child nodes forms a so-called bucket group<sup>1</sup> (see fig. 9). ACK and retransmission processing is restricted to bucket groups.

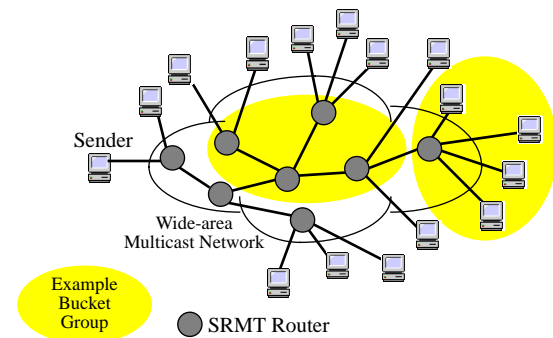


Figure 9. Logical SRMT multicast tree

An additional new feature of SRMT is the use of a distributed bucket algorithm for ACK handling and an XOR combination of several packets for error recovery. The bucket algorithm was originally developed for semi-reliable multicast transmission in XTP (see XTP protocol definition (13)) and in a modified version used for AMTP. In SRMT the bucket algorithm is not performed for the whole group but for each bucket group separately. Buckets are used by an SRMT sender for buffering a fixed number of packets (all of the same size) for possible retransmissions. When the bucket's contents has been transmitted the SRMT sender asks all its SRMT receivers to acknowledge the whole bucket. After a timeout the sender evaluates the ACKs received and starts multicasting selective retransmissions. This procedure is repeated until all SRMT receivers have successfully received all packets of the bucket. Afterwards the SRMT sender deletes the contents of the bucket, which then can be reused for further transmissions. To increase the throughput an SRMT sender may use several buckets in parallel. It is in the responsibility of the receivers to order the packets into the correct sequence before delivering them to the next host or to the application.

For a further reduction of retransmission load an SRMT sender uses XOR combinations of several packets. This

<sup>1</sup> The name bucket is derived from the bucket algorithm used by SRMT for ACKs and retransmissions within a group. These mechanisms will be explained later in this chapter.

mechanism is based on Aghadavoodi Jolfai et al (14) and is best illustrated in an example.

Let a bucket group consist of a sender and three receivers. During transmission of a bucket a single packet is lost for each receiver (say receiver  $i$  misses packet number  $i$ ). Instead of retransmitting all three packets the sender just XORs the bits of the packets and multicasts the resulting packet  $P = P_1 \oplus P_2 \oplus P_3$ . With this single packet each receiver is able to reconstruct its missing packet by XORing  $P$  with the two other packets contained in  $P$ , e.g. receiver 1 gets packet  $P_1$  by computing  $P_1 = P \oplus P_2 \oplus P_3$ .

The set of packets that have to be XORed for an arbitrary error pattern<sup>2</sup> can be pre-computed and stored in tables to avoid high computational cost during transmission.

## 6 Conclusions

Reliable multicast transmission is an essential requirement for many new applications. In this paper a short analysis has illustrated the impact of distribution and size of the multicast group on the performance of existing large-scale reliable multicast protocols. For an improvement we have proposed the SRMT protocol, which permits router-originated retransmissions for packet loss recovery. We believe that the saved transmission cost of SRMT outweighs the cost for the extension of routers. A further traffic reduction can be achieved by the use of a distributed bucket algorithm in combination with XORed retransmissions.

We are currently working on a first implementation of SRMT based on the IP multicast service. Moreover, a more detailed analytical model of different existing protocols and SRMT is under development.

## References

- (1) Deering S., 1988, "Multicast Routing in Internetworks and Extended LANs", Proc. ACM SIGCOMM '88
- (2) Eriksson H., 1994, "MBONE: The Multicast Backbone", Comm. ACM, Vol. 37, No. 8, pp 54-61
- (3) Hermanns, O., Schuba, M., 1996, "Performance Investigations of the IP Multicast Protocols", Comp. Net. & ISDN Sys. 28, pp 429-439
- (4) Armstrong S., Freier A., Marzullo K., 1992, "Multicast Transport Protocol", RFC 1301
- (5) Whetten B., Montgomery T., Kaplan S., 1995, "A High Performance Totally Ordered Multicast Protocol". LNCS, No. 938
- (6) Heinrichs B., 1994, "AMTP: Towards a High Performance and Configurable Multipeer Transfer Service", in "Architecture and Protocols for High-Speed Networks", Danthine, Effelsberg, Spaniol (Eds.), Kluwer Academic Publishers
- (7) Floyd S., Jacobson V., McCanne S., Liu C.-G., Zhang L., 1995, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", Proc. ACM SIGCOMM '95
- (8) Hofmann M., 1996, "A Generic Concept for Large-Scale Multicast", LNCS, No. 1044, pp. 95-106
- (9) Paul S., Sabnani K.K., Lin J.C.-H., Bhattacharyya S., 1997, "Reliable Multicast Transport Protocol (RMTP)", IEEE JSAC, Vol. 15, No. 3, pp 407-421
- (10) Yavatkar R., Griffioen J., Sudan M., 1995, "A Reliable Dissemination Protocol for Interactive Collaborative Applications", ACM Multimedia '95, pp. 333-44
- (11) Kasera S. K., Kurose J., Towsley D., 1997, "Scalable Reliable Multicasting Using Multiple Multicast Groups", Proc. SIGMETRICS '97
- (12) Pingali S., Towsley D., Kurose F., 1994, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", Proc. SIGMETRICS '94
- (13) 1992, "XTP Protocol Definition", Revision 3.6, Protocol Engine Incorporated
- (14) Aghadavoodi Jolfai M., Martin S. C., Mattfeldt J., 1993, "A New Efficient Selective Repeat Protocol for Point-To-Multipoint Communication", Proc. ICC '93, pp 1113-1117

---

<sup>2</sup> The task to determine a minimum set is presumably NP-complete (see again (14)). Nevertheless an exact calculation is possible if the number of packets per bucket is limited.