A Generic Concept for Large-Scale Multicast

Markus Hofmann

Institute of Telematics, University of Karlsruhe,
Zirkel 2, 76128 Karlsruhe, Germany
Phone: +49 721 608 3982, Fax: +49 721 388097
E-mail: hofmann@telematik.informatik.uni-karlsruhe.de
URL: http://www.telematik.informatik.uni-karlsruhe.de/~hofmann[*]

Abstract. Upcoming broadband networks offer a bearer service suitable for modern distributed applications. High bandwidth capacity and low transfer delay are two characteristics of such communication services. Beside these performance-oriented parameters, many applications have additional requirements in respect to quality of the used communication service. Computer Supported Cooperative Work, distributed parallel processing and virtual shared memory, for example, depend on error-free data exchange among multiple computer systems. Additional problems occur for the provision of a reliable multipoint service, where errors are more likely and the sender has to deal with numerous receivers. In order to meet the required reliability, powerful and scalable error control mechanisms are essential. Therefore, this paper presents a novel concept, named *Local Group Concept (LGC)*, for large-scale reliable multicast.

Keywords. Reliable Multicast, Large-Scale Networks, Local Groups, Implosion Control, Retransmissions, Dynamic Groups

1 Motivation

In the near future, global information exchange will become an essential resource in worldwide economies. Forthcoming computer applications will require reliable data transfer within large groups, whose members may be spread worldwide. To support reliable information exchange in such a scenario, protocols have to be changed to provide efficient and scalable multicast error control and traffic control.

One problem new protocols have to deal with is known as the *implosion problem*. As the number of communication participants becomes very large, a sender is swamped with return messages from its receivers. These messages may be generated as a result of status requests or as a result of data loss in conjunction with retransmission-based error control. The effect of implosion is twofold. Firstly, the large number of return messages results in processing overhead at the sender and, therefore, delays data communication. Secondly, a tremendous amount of control units may cause an excess of both, bandwidth and bufferspace, which in turn causes additional message losses. An optimal control scheme for multicast communication would reduce the number of status reports received by the sender down to one.

A second issue of great importance is the development of efficient *error correction* schemes for multicast communication. Common protocols use Go-Back-N or selective

---

[*] In *Proceedings of International Zurich Seminar on Digital Communications (IZS'96)*, Zurich, Switzerland, Springer Verlag, February 1996.

repeat to retransmit lost and corrupted data. Receivers do request missed data directly from the multicast sender without any consideration of network topology and actual network load. In case of group communication, it is also possible to exchange data with other receivers. It is preferable to request lost and corrupted data from a communication participant placed next to the end-system missing some information. An optimal error correction scheme would stimulate the retransmission of missed data units by the receiver located closest to the failing system. This would minimize transfer delay and network load. However, it seems to be very hard to develop an optimal control and error correction scheme. There will be always some kind of trade-off between complexity of protocol mechanisms and benefits to be achieved by an optimal solution.

Many protocols have been developed to support data exchange among various communication participants. Some of these protocols, such as MTP (Multicast Transport Protocol) [1] and RMP (Reliable Multicast Protocol) [2], implement a centralized control and error correction scheme to provide a totally ordered multicast delivery. The central instance controlling data transfer may become a bottleneck when dealing with numerous receivers. Therefore, these protocols will not scale well in respect of the group size. An approach based on the establishment of so called multicast servers within the network is given in [3]. Other protocols with centralized, sender-based control integrate special mechanisms to avoid acknowledgment implosion. A recent version of XTP (Xpress Transfer Protocol) [4], for example, defines damping and slotting algorithms to reduce implosion. Instead of returning control units exclusively to the multicast sender, receivers transmit them after a random delay to the whole group. Consequently, every group member receives this message and skips its status report if the incoming control unit corresponds to its personal state. This mechanism might reduce the number of acknowledgments to be processed by the sender. But in large-scale global networks, the usefulness of multicasting control units is very questionable [5]. A large number of participants together with the mechanism described above may cause a flood of control units all over the global network. SRM (Scalable Reliable Multicast) [6], which integrates a similar mechanism to implement a receiver-based error control, uses timers carefully set to avoid a flood of retransmission requests. However, the correct setting of timers will be very difficult for high dynamic networks with quickly changing load and frequently changing network structure.

This paper presents a novel error control mechanism suitable for global heterogeneous networks. It combines the advantages of both, sender-based and receiver-based control schemes. The concept aims at reducing transfer delay, network load and the acknowledgment processing overhead at the sending side. Section 2 and Section 3 present the proposed framework based on the Local Group Concept (LGC). Section 4 gives results of a performance analysis in respect to a simple example scenario. Finally, Section 5 draws concluding remarks.

2 The Local Group Concept (LGC)

The Local Group Concept (LGC) presented in this paper is based on a best-effort delivery model with multicast support. While these requirements are perfectly in conformity with IP and the current MBone, the Local Group Concept is not restricted to the Internet protocol family. Although this paper focuses on the use of the traditional IP multicast distribution model, the generic concept can also be integrated in an extended ATM Adaption Layer or in other protocol architectures with multicast support. Additionally, LGC avoids changes within internal network equipment, such as ATM switches or IP routers. The integration of implosion control into these systems requires protocol processing capabilities inside themselves. This implies more complex and less flexible intermediate systems. In addition, a router doing implosion control for hundreds of multicast connections will soon become a bottleneck. Therefore, the LGC approach focuses on implosion control on end-to-end basis.

2.1 Basic Principles of LGC

The problem with implosion is that a single system has to control a very large number of receivers. The basic principle of the Local Group Concept is to split the burden of acknowledgment handling and to distribute error correction as well as traffic control among all the members of the multicast group. To achieve better scalability of point-to-multipoint services in respect of group sizes, distances, and data rates, LGC divides global multicast groups into separate subgroups. These subgroups will combine communication participants within a local region, forming so called *Local Groups (LG)*. Each of them is represented by a specific communication node, named local *Group Controller (GC)*. These nodes support the provision of the following enhanced services:

- *Local Retransmissions:* Controllers of Local Groups are able to perform and to coordinate retransmissions of lost and corrupted data within their subgroup. This reduces delay caused by retransmissions and decreases the load for sender and network.
- *Local Acknowledgment Processing:* The integration of acknowledgment processing capabilities into local Group Controllers reduces the acknowledgment implosion problem of reliable multicast for large groups. Local Group Controllers evaluate received control units and inform the multicast sender about the status of the Local Group. This includes error reports as well as parameters to control data flow. Parallel processing of status reports and their combination to a single message per Local Group relieves the multicast sender as it reduces the number of control units to evaluate at the sending side.

In each local region, one of the receivers is determined to function as local Group Controller. The sender itself is defined always as a local Group Controller. The dedicated system has to collect status messages from all members of its subgroup and has to forward them to the multicast sender in a single composite control unit. Controllers of subgroups are also responsible for organizing local retransmissions. After evaluating received status messages a local Group Controller tries to transfer lost data units to all receivers that have observed errors or losses. To retransmit data units a local Group Controller can use either unicast or restricted multicast transmission. This decision may be static or dynamically based on the number of failed receivers. If a controller itself misses some data units, it will try to get them from another member of its Local Group. Therefore, a multicast sender has only to retransmit messages missed by all members of a subgroup. Local retransmissions lead to shorter delays and decrease the number of data units flowing through the global network.

An example scenario illustrating the basic idea and the advantage of the Local Group Concept is given in Figure 1. A multicast sender communicates over a satellite link with four receivers, which are connected to a common router. The satellite link is characterized by high transfer delay and high carrier fees. Therefore, it is desirable to reduce data traffic over this link. In this type of scenario it is useful to combine all four receivers into a single subgroup. One of the receiving end-systems has to function as the controller of the subgroup. In this case, local retransmissions do not traverse the satellite link. This reduces transfer delay and network load within the satellite link.

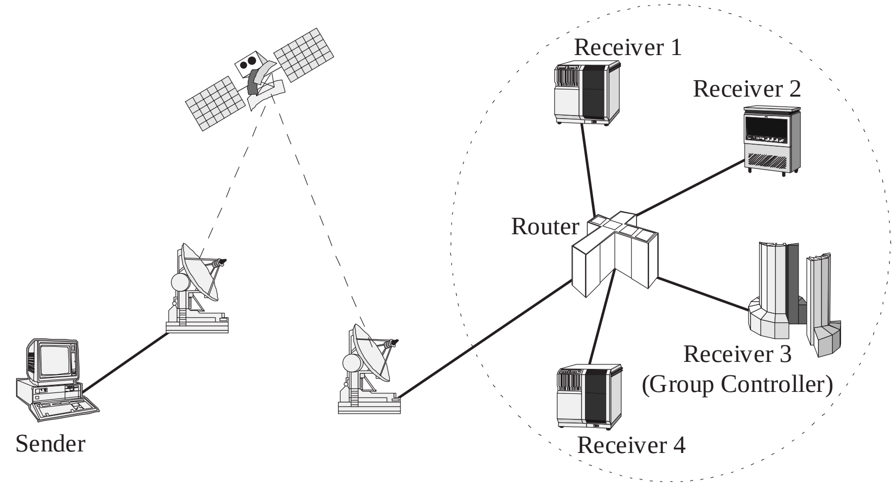

Fig. 1. Combining End-Systems to one Local Group

In this example, the decision to combine all four receivers into one single subgroup was based on the intention of minimizing transfer delay. The next Section deals with aspects that influence the creation and structure of Local Groups.

2.2 Metrics for Building Local Groups

The Local Group Concept is based on the combination of receivers within a 'local region'. Thus, the term 'local region' has to be clarified. Some kind of metric is necessary to determine the distance between two communication nodes. The suitability of different metrics, such as delay, bandwidth, throughput, error probability, reliability, carrier fees, or number of hops between two nodes, depends mainly on the application using a communication service. While an interactive application may wish to minimize transfer delay, a user transferring data files is interested in reducing financial costs of the transfer. However, there is a trade-off between complexity due to general distance metrics and efficiency when using simple metrics such as hop counts. A discussion about the usefulness of metrics such as transfer delay or hop counts can be found in [7] and [8]. The decision which kind of metric can be used also depends on the underlying network protocol. In some cases it is necessary that the routing protocol supports the chosen unit of measurement. IP, for example, considers metrics like delay, throughput, and reliability, while the ISO protocol IS-IS supports bandwidth, error probability, delay and financial costs [9]. Combinations of these mentioned metrics can also be taken into consideration. If the underlying network service supports no metric at all, the measured round trip time may be defined as the distance between two stations.

A simple solution for a similar problem has been proposed in [10]. The generation of local multicast regions as defined in the Designated Status Protocol (DSP) and the Combined Protocol (CP) is mainly based on addresses and the geographical location of end-systems (for example, end-systems connected to a common switch). Such a static mechanism leads to problems supporting huge MANs (e.g., DQDB or FDDI) and mobile environments. The definition of local regions based on the geographic location of end-systems does not work very well in general, as mentioned in [7], because geographic information often does not correlate to network structure. Therefore, the Local Group Concept aims at flexible, dynamic and application-depending generation of Local Groups.

2.3 Selection and Placement of Group Controllers

An important detail in the design of a concept like LGC is the decision where to place the functionality of local Group Controllers. The question arises which type of communication system is suitable to be responsible for controlling a Local Group. Possible answers are, for example, end-systems, routers, or switches. If there are several suitable systems, which selection will fit best?

The CP protocol mentioned in [10] uses local exchanges (e.g., routers or ATM switches) to combine status messages of receivers within a local region. This solution implies modifications of local exchanges and restricts the Local Group Concept to operate in the network layer. By placing local Group Controllers into end-systems, the concept becomes ‘layer-independent’. In addition, the end-system-based approach avoids the necessity of any changes in switches and routers. In this case, the protocol mechanism can be used in networks based on different protocol architectures. The placement of local Group Controllers in end-systems scales better for a large number of multicast connections. A local exchange doing retransmissions and acknowledgment processing for different multicast connections may soon become a bottleneck. It is advantageous to parallelize the job of handling different Local Groups and to distribute it to several communication nodes. Each multicast connection should have its own, separate local Group Controller. The end-system based solution allows for choosing different communication participants within the same local region for different multicast connections.

The Local Group Concept presented in this paper integrates the functionality of local Group Controllers into normal end-systems. In principle, every communication participant is able to perform the tasks necessary for controlling a subgroup. This strategy is similar to the selection of a monitor station within a Token Ring network [11]. One member per Local Group is designated as the active controller. The selected station represents the Local Group and performs local acknowledgment processing and local retransmissions.

Another important issue is the dynamic change of communication groups and network traffic. The structure of Local Groups and the placement of their controllers should always be adapted to the actual state of the network and the communication group. Therefore, it must be possible to move functionality of local Group Controllers during the lifetime of a connection.

The DSP protocol described in [10], for example, combines status messages in end-systems, which are selected at connection setup time. For that reason, no chance to reassign the role of a Group Controller is given. Careful selection and placement of local Group Controllers assist in optimizing network load and transfer delay. Characteristics such as processing capacity, memory size, and location within the network, determine the suitability of an end-system for the role as a local Group Controller. Other important issues concern network topology, group size, and group structure. Due to this wide range of parameters, there is no simple algorithm to find the optimal local Group Controller. Instead, near-optimal algorithms or other heuristics could be applied. A simple solution is to define the member next to the multicast sender as local Group Controller.

LGC defines local Group Controllers in the following manner. The initiator of a new Local Group always becomes the first controller of this subgroup. After a certain period of time or after joining of a new member, the local Group Controller initiates a reconfiguration process. During this process, every member of the Local Group reports its parameters relevant for the decision to the local Group Controller. Like the metric used to determine the distance between two systems, these values may depend on the application using the communication service. Possible parameters are distance to the multicast sender, performance, and buffer space of each end-system or the location of a receiver in relation to other members of the Local Group. Based on information obtained from its members, the local Group Controller determines the most suitable end-system to take over the role as local Group Controller in the future. If a change is advantageous, all necessary information migrates from the old to the new controller and all members of the Local Group are informed about this change. During this reconfiguration process, every join or leave request is queued and performed later.

2.4 Hierarchy of Local Group Controllers

To improve the benefits of LGC, Local Groups can be organized in a hierarchical structure. A hierarchy of Local Groups exists, if a local Group Controller LG_2 does not report directly to the multicast sender. Instead, it sends retransmission requests and status reports representing the whole Local Group to a local Group Controller LG_1 next to itself. This one treats LG_2 like a normal receiver. Therefore, all retransmission requests of LG_2 are redirected to and satisfied by LG_1. In this case, LG_1 is named the *parent* of LG_2, while LG_2 is the *child* of LG_1. It is obvious that this hierarchy will decrease transfer delay, network load, and implosion additionally.

Two changes have to be made to support a hierarchical structure. Firstly, every local Group Controller (except the multicast sender) has to subscribe to the nearest Local Group as a normal receiver. Secondly, every local Group Controller has to redirect its control messages to the next local Group Controller at an upper level (to its parent). Additionally, it may be necessary to increase timeout values of parents, because children have to collect all status reports of their members before answering to their parent. This timeout adaption could be done dynamically during a synchronizing handshake. The remainder of this paper does not consider the hierarchy in more detail. Nevertheless, all the algorithms described in the following Sections may be easily modified to support hierarchical Local Groups.

3 Protocol Mechanisms realizing LGC

The following Subsections give a more detailed description of the Local Group Concept. A complete specification including pseudo code for integrating the concept into existing multicast protocols can be found in [12].

3.1 Joining Group Communication

Before receiving any data, new communication participants have to subscribe to a nearby Local Group in respect to the chosen metric. A simple possibility to find an appropriate local Group Controller is to use a service similar to the Host Anycasting Service described in [13]. The search may be based on routing information, too, obtained directly from routing tables. If there is no support by the network layer at all, an expanding ring mechanism [7] based on round trip times can be used. The overhead produced by this search strongly depends on the mechanism applied to find an appropriate Local Group. An end-system using the Host Anycasting service just sends one anycast request and gets back a single answer within a period of time corresponding to the round trip time between itself and the answering Group Controller. In contrast, an expanding ring search will cause much more overhead. Depending on the result of the search, different actions have to be performed:

- If no Local Group is found within a defined distance according to the chosen metric, the joining end-system establishes a new subgroup, appoints itself as Group Controller and informs the multicast sender about this event. Hence, the multicast sender has knowledge of all Local Groups and their representative controllers.
- If an appropriate Local Group exists within a certain distance, the joining end-system registers itself at the corresponding Group Controller. Therefore, a local Group Controller always knows the identity of all its members. This is a prerequisite for providing a reliable multicast service.

After subscription to a Local Group, the end-system is allowed to participate in the group communication. The division of dynamic communication groups into local subgroups has to be updated after a certain time. Due to dynamic changes in group membership, it is necessary to modify the group structure. For example, it may be preferable to place the local Group Controller in a newly joined participant that is located in the center of the subgroup. Such a reconfiguration process could be implemented according to the mechanism mentioned in Subsection 2.3. A reconfiguration is also necessary, if the division into Local Groups is based on routing information, which can be incorrect for limited periods of time and which is not globally consistent. Periodic updates of the group structure correct errors caused by temporary inconsistent metric information.

Due to the knowledge of group membership, a multicast sender is able to guarantee the correct data transfer to all of its children. Local Group Controllers are responsible for correct transmission of user data to all members of their subgroup. This allows to provide a reliable multicast service that scales well for either long distances and a large number of receivers.

3.2 Transfer of Multicast Data

The sender multicasts data units (e.g., using IP-multicast) to all destinations. After a certain period of time, the sending station requests the status of all receivers. Instead of returning control units directly to the sender, receivers transmit them to the controller of their Local Group. The controller collects incoming status messages, processes them, and calculates the status of the whole subgroup. Applying the knowledge of received control units, a local Group Controller requests data units missed by itself from other members of the subgroup and performs all relevant local retransmissions. In addition, the controller combines received status messages into a single acknowledgment and transmits it to the multicast sender. The acknowledgment includes the request for all data units missed by all members of the Local Group. This mechanism ensures that every data unit received successfully by at least one group member is not requested at the multicast sender anymore, even if the Group Controller itself misses this data unit.

After the reception of user data, receivers deliver the information to the service user. However, they are not allowed to release the corresponding buffer space. This is because the local Group Controller may miss these data and request it at a later stage. Therefore, receivers have to wait either for a sufficient amount of time, or for an explicit permission by the local Group Controller before releasing the receive buffer. The period of time data must be kept by receivers depends among other issues on the radius of the Local Group and the frequency of status requests.

3.3 Leaving Group Communication

If a normal receiver wants to leave group communication, it unsubscribes itself at the corresponding local Group Controller. After performing necessary retransmissions, the Group Controller confirms unsubscription and the receiver is allowed to leave.

If an end-system functioning as a local Group Controller wants to leave communication, first of all an appropriate successor has to be determined. The new controller obtains all relevant group management information, like status and member information, from its predecessor. Afterwards, all members of the Local Group are informed about the identity of their new controller. Henceforth, it is the new local Group Controller that deals with new data sent by the multicast sender and that receives status reports from group members. However, the leaving Group Controller has to ensure immediately that all outstanding retransmissions to its former members have been dealt with, before being allowed to leave.

If the leaving end-system is the only one within its Local Group, it just unregisters itself directly at the multicast sender or at its parent.

3.4 Fault Tolerance

With respect to fault tolerance it is interesting in which way the Local Group Concept handles failures of a local Group Controller. The question arises, how sending and receiving end-systems discover such a failure and what will be done in this case.

As mentioned above, local Group Controllers must periodically send messages to initiate a buffer release at receiving sides. If a local Group Controller fails, receivers will not get such messages any more. In this case, they infer a failure of their local Group Controller. The multicast sender or other controllers within a Local Group hierarchy detect a failing child from the missing answer in response to a status request. Every end-system reports such a failure to its parent.

The reaction to the failure of a local Group Controller depends strongly on the application given group semantic. If the application requires an all reliable communication service and at least one of the local Group Controllers fails, the connection will be aborted and an error will be indicated to the service user. If the failure of a receiver is acceptable and an active controller fails, another group member has to take its role. For example, this could be a replicated local Group Controller (similar to the solution presented in [14]) or a dynamically determined successor within the Local Group. It is also possible that all members of the failed Local Group search for a new Local Group according to the mechanism described in Section 3.1.

3.5 Integration of Local Groups in XTP

As an example of integrating the novel concept into existing transport protocols, the mechanisms described above have been adopted to XTP, version 3.6 [4]. Work is ongoing to integrate the Local Group Concept into an implementation of XTP 4.0 [15]. An additional packet type for migration of management information must be added. To implement a k-reliable multicast service, sender and Group Controllers must be able to distinguish between different receivers. Therefore, an additional field identifying individual source addresses has to be integrated into CNTL packets of XTP.

4 Performance Evaluation

Analytical methods were applied and simulations were performed in order to evaluate the benefits of the Local Group Concept. The simulations were based on BONeS/Designer, an event-driven network simulation tool by Comdisco. During evaluation, two important measures were focused:

- *Transfer Delay*, which is the primary concern from the application's point of view.
- *Network Load*, the most interesting measure from the networker's perspective.

The simulation scenario consists of a sender and various receivers that are connected to a local area network. The sender is linked to the LAN over a wide area network. If the satellite link in Figure 1 is replaced by a wide area network, these two scenarios will be identical.

The evaluation examined the impact of group size on average transfer delay and network load. The values obtained for the Local Group Concept were compared to the corresponding results for two common sender-based techniques using multicast and unicast retransmission. Transfer delay was assumed to 20 ms for the wide area link, which is approximately the delay for transferring data between the East and West coast of the USA. Transfer delay within the local area network was fixed at 2 ms. Error probability was assumed to be 10^{-1} for message loss caused by buffer overflow or bit errors, which is not uncommon for highly loaded internetworks. This value was derived from many measurings to randomly selected internet hosts using *ping* command. Other simulation parameters included data rate, processing delay within communication systems, status request rate, and burst length.

The impact of group size on average transfer delay is given in Figure 2. The graph shows for all techniques an increase in the average transfer delay with increasing numbers of receivers. The sharp increase for the common, sender-based approach can be explained with the large number of acknowledgments that have to be processed solely by the multicast sender. The relative high values for average transfer delay are mainly caused by retransmissions over the wide area link. Local retransmissions, as proposed by the Local Group Concept, effect a less strong increase in transfer delay. The benefit for the Local Group Concept is extremely high for large communication groups, but transfer delay still increases with the number of receivers. The establishment of several Local Groups with a restricted number of receivers leads to a distribution of acknowledgment processing to different local Group Controllers. In the third simulation, the maximum size of Local Groups was defined to 50 members. Therefore, every Group Controller just has to process a maximum of 50 acknowledgments. This results in an soft upper bound for transfer delay, because every Local Group with more than 50 receivers is separated into diverse subgroups with fewer members. This division distributes the burden of acknowledgment processing to different controllers and avoids further increase of transfer delay due to acknowledgment processing overhead.

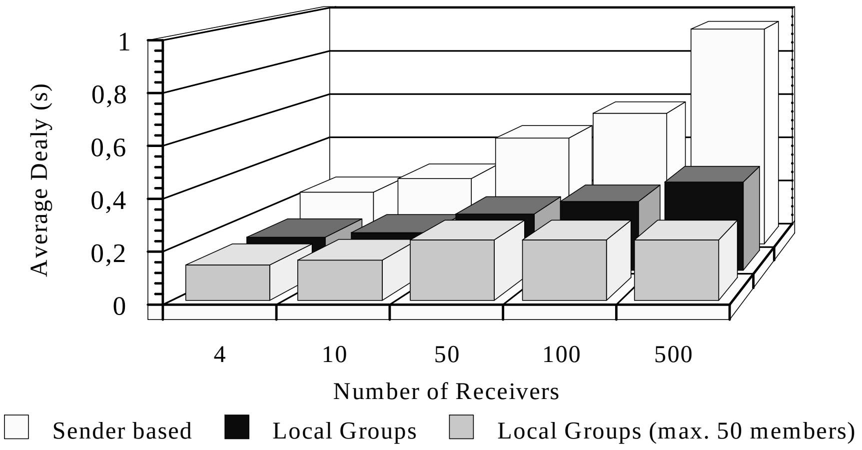

Fig. 2. Impact of Group Size on Transfer Delay

To clarify the difference between unicast and multicast retransmission for both of the common techniques, 50 receivers were added at the sending side of the wide area network for evaluation of network load. The result is illustrated in Figure 3.

For both common techniques, the ratio of retransmissions and total data traffic over the wide area link depends directly on the number of recipients. The Local Group Concept, in contrast, is not influenced by the number of participants and shows very low values compared with the results of the other techniques. The independence between network load and the number of receivers simplifies resource reservation for highly dynamic communication groups. The bandwidth required on the wide area link can be reserved regardless of the actual or future group size. Additional resource reservation during the data transfer phase due to joining participants is not required.

The Local Group Concept requires additional memory at receiving sides due to the delayed release of receive buffers. Simulations have demonstrated that the maximum amount of additional buffer space is independent of the error probability and of the group size. It corresponds directly to the number of data units received between two succeeding release messages. If every tenth data unit is followed by a release message from the corresponding Group Controller, the additional memory at receiving sides will not exceed the buffer space required to save ten data units. Therefore, the need for additional memory can be regulated by the frequency of release messages.

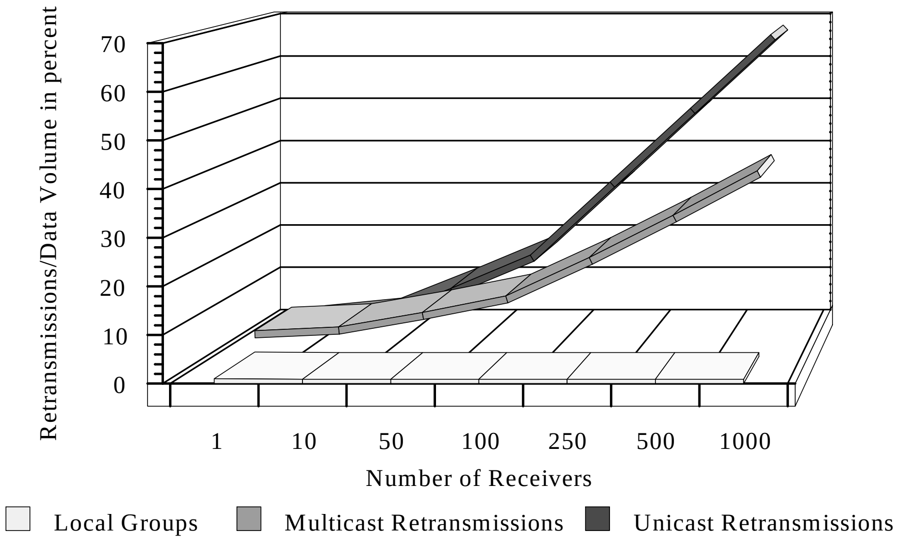

Fig. 3. Impact of Group Size on Network Load

Additional results and detailed information about the analytical methods, the simulation tools and the simulation models can be found in [12].

5 Conclusion

The general goal of the developed concept is the provision of scalable protocol mechanisms appropriate for multicast communication in global heterogeneous networks. Performance evaluation including simulation and analysis of network load shows for the Local Group Concept significant improvements compared to common multicast techniques. The benefits arise from replacing global retransmissions by local data exchange and additionally, from parallel preprocessing of acknowledgments within Local Groups. The performance benefits are achieved without the necessity to modify network equipment such as ATM switches or IP routers, and without a dramatically increased buffer overhead.

Careful placement of local Group Controllers assists in optimizing transfer delay and network load. Depending on particular group dynamics and network topology, it is advantageous to place controllers near or at the centre of a Local Group. There is no simple algorithm to find the centre of a dynamic group at all. Instead, some kind of heuristic has to be developed defining the right balance between simplicity and optimal placement of the controller.

6 Acknowledgments

I would like to thank my colleagues Torsten Braun and Georg Carle for valuable comments and suggestions on various subjects of this paper. In addition, thanks to Ralf Seufert, who performed some of the simulations presented in this paper.

References

[1] S. Armstrong, A. Freier, K. Marzullo: *Multicast Transport Protocol*. RFC 1301, February 1992

[2] B. Whetten, T. Montgomery, S. Kaplan: *A High Performance Totally Ordered Multicast Protocol*. Submitted to INFOCOM'95, April 1995

[3] Georg Carle: *Error Control for Reliable Multipoint Communication in ATM Networks*. Third International Conference on Computer Communications and Networks - ICCCN'96, San Francisco, September 1994

[4] W. Strayer, B. Dempsey, A. Weaver: *The Xpress Transfer Protocol*. Addison-Wesley Publishing Company, 1992

[5] B. Heinrichs, K. Jakobs, A. Carone: *High performance transfer services to support multimedia group communications*. Computer Communications, Vol. 16, No. 9, September 1993

[6] S. Floyd, V. Jacobson, S. McCanne, C. Liu, L. Zhang: *A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing.* Computer Communication Review, Vol. 25, No. 4, Proceedings of ACM SIGCOMM'95, Cambridge, Massachusetts, August 1995

[7] J. Guyton, M. Schwartz: *Locating Nearby Copies of Replicated Internet Servers.* Computer Communication Review, Vol. 25, No. 4, Proceedings of ACM SIGCOMM'95, Cambridge, Massachusetts, August 1995

[8] M. Crovella, R. Carter: *Dynamic Server Selection in the Internet*. Third IEEE Workshop on High Performance Subsystems (HPCS'95), Mystic, Connecticut, August 1995

[9] Radia Perlmann: *Interconnections: Bridges and Routers*. Addison-Wesley Publishing Company, 1992

[10] S. Paul, K. Sabnani, D. Kristol: *Multicast Transport Protocols for High Speed Networks*. Proceedings of International Conference on Network Protocols, Boston, October 1994

[11] W. Stallings: *Local and Metropolitan Area Networks*. Fourth Edition, Macmillan Publishing Company, 1993.

[12] Markus Hofmann: *Reliable Group Communication in Global Heterogeneous Networks*. Diploma Thesis at Institute of Telematics, University of Karlsruhe, August 1994 (in german)

[13] C. Partridge, T. Mendez, W. Milliken: *Host Anycasting Service*. RFC 1546, November 1993

[14] H. Holbrook, S. Singhal, D. Cheriton: *Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation*. Computer Communication Review, Vol. 25, No. 4, Proceedings of ACM SIGCOMM'95, Cambridge, Massachusetts, August 1995

[15] XTP Forum: *Xpress Transport Protocol Specification, XTP Revision 4.0*. XTP Forum, Santa Barbara, CA, USA, March 1995