

Scalable Multicast Communication in the Internet

by Markus Hofmann, University of Karlsruhe

Introduction

Driven by the capabilities of modern high speed networks, a new generation of distributed systems is emerging. These systems make the support of superior distributed applications technically feasible, while increasing demand on communication in distributed environments has made them necessary. Forthcoming applications, such as collaborative distributed work, videoconferencing, or information dissemination, are expected to require information exchange between a large number of geographically dispersed components. They typically make use of a specific form of communication in which a single sender transmits data to multiple receivers. This form of communication is called *multicast* communication. This article shortly presents the basic principles of efficiently providing multicast services, discusses the problem of scalability with respect to group size and presents recent research approaches to overcome existing bottlenecks.

Benefits of multicast

Multicast data transfer could be realized by repeatedly transmitting data units using point-to-point transfer to every communication participant. However, this approach does not scale well with the number of recipients and increases network load by the reciprocal of the group size. On the other hand, broadcasting data units may be an acceptable solution for small networks, but it causes a flood of data packets in wide-area networks. Rather than broadcasting information or using multiple unicast transfers, the forward-looking approach is to form a multicast group consisting of an arbitrary number of receivers and a single transmitter. Every data unit sent to the multicast group will be delivered only to those hosts that have registered themselves as member of the group. This requires special capabilities and additional mechanisms at different protocol levels.

Link level multicast

Protocol architectures designed to support multicast communication efficiently include special mechanisms even on the data link layer. Every data packet received by a network interface causes an interrupt and stimulates further processing at higher protocol levels. Therefore, it is desirable that hosts receive and process only those packets which are destined to them. Multicast protocols meet this desire by taking advantage of address filtering on the data link layer. Multicast capable interfaces can be configured to accept and process multicast packets in addition to directly addressed data frames. This renders possible the use of multicast addresses instead of the "all hosts" address or multiple unicast addresses. Transmitters connected to a broadcast media (e.g. Ethernet) need to send only one copy per packet without the consequence of causing further packet processing at non-member hosts.

Network level multicast

While transmitting one copy per packet is sufficient in local environments based on broadcast media, it is necessary to send multiple copies across ATM based networks or across heterogeneous internetworks (e.g. the Internet). In such a scenario, transfer of multicast messages can be optimized by delaying the replication of a data packet until it has to

traverse different links. Therefore, routers and switches have to incorporate group management facilities as well as mechanisms to establish and maintain multicast routing trees. To enable early usage of multicast services without waiting for the availability of complete standards and without the need for wide-spread use of multicast-capable routers, the *Multicast Backbone (Mbone)* [1] has been established. The Mbone is an overlay network on top of today's Internet providing a multicast facility to the network community. It makes use of Deering's IP multicast extensions [2] and of multicast capable routing protocols, such as the *Distance Vector Multicast Routing Protocol (DVMRP)* [3] or the *Multicast Open Shortest Path First (MOSPF)* [4]. However, research in the field of multicast routing is still going on [5]. New mechanisms are necessary to avoid explosion of states for wide-area routing and to support policy routing. A promising approach is the *Core Based Tree (CBT)* strategy [6], which reduces state information necessary to be kept within routers down to one per group.

Today's Mbone comprises only a small fraction of currently installed Internet routers and uses so called *tunnels* to link the multicast-capable islands together. These tunnels are manually configured by system administrators. They are used to forward multicast packets through non-multicast routers by encapsulating them inside regular IP packets. The Mbone has been established to get experience with the new multicast technology. However, the recent success of applications deployed over the Mbone illustrates the enormous potential of group communication and demonstrates the instant need for multicast services in wide-area networks.

Transport level multicast

Multicast capable networks provide efficient and scalable routing of data packets to multiple receivers. However, the bearer service provided by these communication networks does not fit the requirements of some applications. IP multicast, for example, offers an unreliable datagram service. The provision of reliable data transfer requires additional protocol mechanisms. According to the Internet protocol architecture, reliability as well as flow and rate control should be provided on an end-to-end basis. Therefore, mechanisms to support reliable multicast delivery should be integrated in transport level protocols.

When designing multicast protocols, scalability becomes more and more important. Widespread availability of IP multicast and development of applications deployed in the Mbone have considerably increased the geographic extent and the size of communication groups. Extensive use of services like Internet radio or large-scale conferencing leads to thousands of receivers being involved in a single multicast communication. In addition, communication participants may be spread all over the world. As the size and the geographic span of communication groups increases, efficient connection management schemes including scalable error and traffic control become more and more essential. Recent research projects deployed new transport level protocols to meet the requirements of large-scale multicast applications in heterogeneous networks. Some proposals only address particular aspects of group com-

munication and focus only on some specific user environments. However, multimedia applications often have to handle several highly diverse data streams at the same time. A system supporting distributed cooperative work, for example, might offer shared use of a text editor while simultaneously providing audio and video communication between all the participants. Such applications require a multipurpose and flexible communication subsystem. On the other hand, other proposals suffer from a high degree of complexity and are too general in some issues. However, reliable data delivery is required by a wide range of applications. Therefore, the article will focus on protocol mechanisms for the provision of a reliable multicast service.

Reliable multicast

Measurements have shown, that packet losses in the current Mbone are significant. The data sets collected in [8] state that in one scenario almost 70% of transmitted packets were not successfully received by at least one receiver. This illustrates the need for efficient and powerful error correction schemes.

Some kind of interaction between sender and receivers is necessary to ensure correct data delivery as well as to perform any kind of congestion or flow control. Neither of the hosts has enough information to control data streams on its own. The provision of reliable data transfer, for example, is based on a comparison between sent and received data. The transmitter has knowledge about which data units have been sent and the receivers about which data units have been received successfully. Therefore, the provision of a reliable communication service requires the transmission of receiver status back to the sender or vice versa.

Sender-based error control

So called *sender-based* schemes, in which the transmitter is responsible for controlling data transfer, rely on collecting status information at the sending site. Receivers transmit control units, including acknowledgments and traffic control information, back to the sender. As the number of receivers becomes very large, the multicast sender is overwhelmed with return messages of its receivers. This fact is known as *sender implosion*. The effect of implosion is twofold. Firstly, the large number of return messages results in processing overhead at the sender and, therefore, delays data transfer. Secondly, an enormous amount of control units may cause an excess of both, bandwidth and bufferspace, which in turn causes additional message losses. While the latter issue is essential in wide-area networks and within large communication groups, processing overhead becomes more important in LAN/MAN environments. Over the last years, transmission capacity has been growing immensely, while protocol processing has turned out to be a performance bottleneck. Future photonic networks, for example, will provide bandwidth at no cost, but processing time will still remain a valuable resource. The processing bottleneck is particularly crucial for multicast communication. With an increasing number of receivers, processing of an increasing number of control packets needs to be performed. An optimal control procedure would reduce the number of status reports received by the sender down to one.

Receiver-based error control

Other approaches use *receiver-based* schemes, where receivers themselves are responsible for error detection and error recovery. They need not to return status reports or acknowledgments to the transmitter. Instead, receivers use negative acknowledgments to request missed or corrupted data units from the transmitter. This reduces the number of return messages and prevents sender implosion. However, it is not possible to provide a full-reliable communication service while using such an error recovery scheme. The failure and dropping out of a single receiver could not be detected by the transmitter. Furthermore, flow or rate control always require some kind of status exchange between sender and receivers. This could lead to sender implosion even in the case of receiver-based reliable multicast communication.

Forward error correction

Forward error correction (FEC) has also been proposed to provide a reliable communication service. It is suitable for real-time applications since it allows error recovery without adding any delay associated with retransmissions. However, FEC does not prevent sender implosion caused by messages necessary for traffic control. Moreover, adding redundant control information wastes bandwidth even when no or just a few errors occur. The sender must add enough control information to enable correction of all errors. While receivers in heterogeneous networks have highly diverse error characteristics, it is not adequate to choose a single fixed level of redundant coding. Such a fixed coding level may be excessive for some receivers while it may be insufficient for others. In addition, the error characteristic of receivers may change dynamically due to processing load, buffer occupancy, or network load. Therefore, the coding level should be adapted according to the current receiver state.

Retransmission schemes

Retransmission schemes may be quite effective, even for error control in multicast protocols geared to support real-time applications [9]. Retransmissions of data units are used to close gaps in the data stream of the receivers. Most protocols use Go-Back-N or selective repeat to retransmit lost and corrupted data. Receivers do request missed data units directly from the transmitter without any consideration of network topology and current network load. In the case of group communication, it is also possible to exchange data with neighboring receivers. It is preferable to request lost and corrupted data from a group member placed next to the host which is missing some information. An optimal error correction scheme would stimulate retransmissions of missed data units by the receiver located closest to the failing host. This would minimize transfer delay and network load. Studies of packet loss correlation in the current Mbone [8] show that packet loss is more likely to occur on the path between the multicast backbone and the local host rather than on the backbone links of the multicast tree. The measurements also show that, on average, there is just a little pair-wise spatially associated loss in the Mbone. Therefore, the probability that a receiver is able to get a missed data unit from a nearby group member is quite high.

Local Group Concept

The described characteristics of the Mbone have strongly influenced the design and the development of a novel multicast protocol, which is called the *Local Group Concept (LGC)* [10]. The mechanisms of LGC are designed to support full-reliable and semi-reliable data transfer in large-scale, heterogeneous networks. They are 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. It can also be integrated in an extended ATM Adaptation Layer or in other protocol architectures with multicast support.

Defining Local Groups

The basic principle of LGC is to split the burden of acknowledgment handling and to distribute error correction among all the members of a multicast group. To achieve better scalability of point-to-multipoint services, LGC splits global communication groups into separate subgroups. These subgroups will combine communication participants within a local region, forming so called *Local Groups*. Each of them is represented by a *local Group Controller* that collects status information from the members of its local group. The local Group Controller evaluates these return messages, combines them into a single control packet and transmits it back to the multicast sender or a higher level local Group Controller. Local Group Controllers also support the provision of local retransmissions. They coordinate local recovery from data loss to avoid expensive retransmissions from the multicast sender. This reduces delay and decreases the load for transmitter and network. The integration of message processing capabilities into local Group Controllers reduces the implosion problem of multicast traffic and error control 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 into a single message per Local Group relieves the multicast sender as it reduces the number of control units to be evaluated at the sending side.

In each local group, one of the receivers is determined to function as local Group Controller. The dedicated system has to collect control messages from all the 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 the 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 a data unit, it will ask another group member to multicast this data unit to the 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 a global network.

Example

An example scenario illustrating the basic idea and the advantage of this concept is given in Figure 1. A multicast sender communicates over a satellite link with four receivers, which are connected to a common switch. 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 hosts 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.

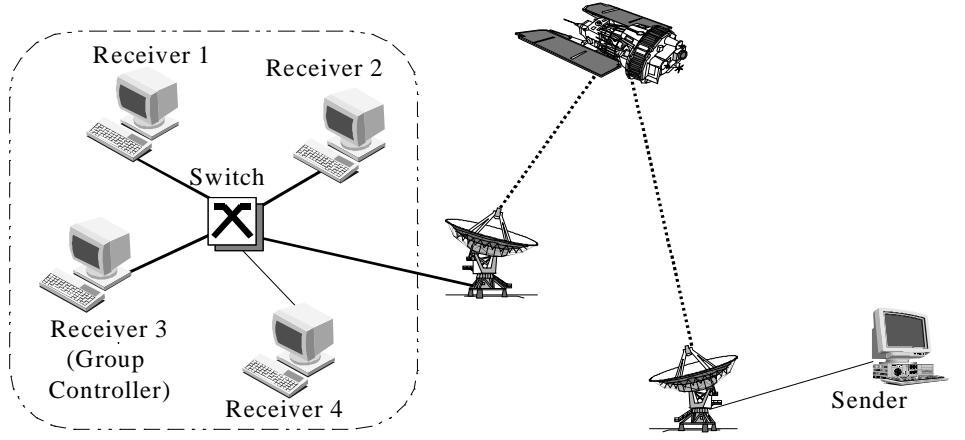


Figure 1: Example for the Definition of Local Groups

The resulting data streams are shown in Figure 2. The transmitter multicasts data units directly to all group members using a multicast capable delivery service (1). The local Group Controllers are kept out of the outgoing data path avoiding an extra handling of data units at each level of the hierarchy. Therefore, local Group Controllers need not to store data fragments, reassemble complete data units, interpret and forward them. Instead, multicast forwarding is done by the delivery service in a more efficient way. After receiving a status request, regular receivers transmit control messages to their corresponding local Group Controller (2). The controller combines the status reports into a single control unit and sends it to the multicast transmitter (3). Therefore, it could be said that the Local Group Concept causes some kind of triangulation of data flow.

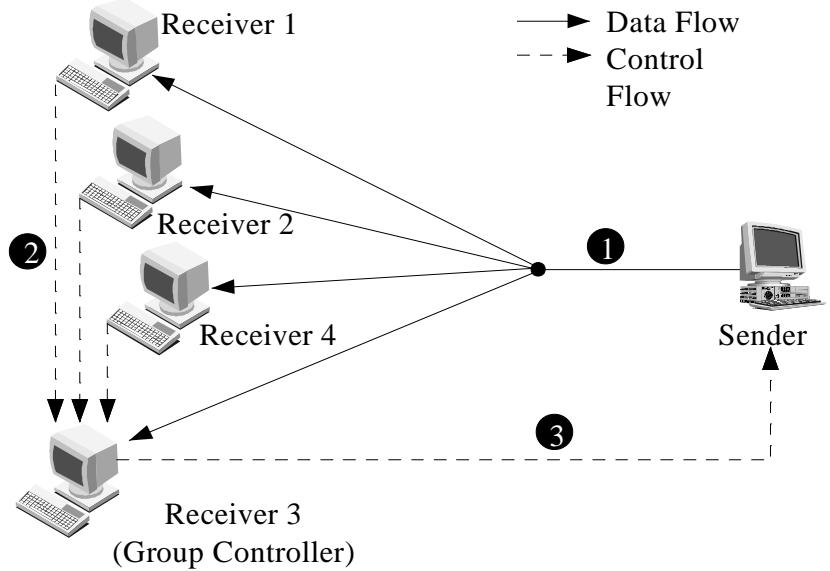


Figure 2: Triangulation of Data Flow

Local Group reformation

The Local Group Concept also includes mechanisms to increase fault tolerance [10]. The Fail Stop of a local Group Controller, for example, is handled by dynamic reformation of Local Groups. Of course, actions performed after the failure of a communication participant depend on the given group semantic. If the communication user requires an all-reliable service, the multicast connection will be closed due to the failure of a group member. In the case of semi-reliable service, the actions to be performed depend on the attributes and the role of the failed receiver. Dynamic group reformation is also used to adapt the group structure to the current network load and to the current state of all communication participants. For further improvements in respect to transfer delay and implosion, Local Groups are organized in a hierarchical structure.

Metrics

In the above example, the decision to combine all four receivers into one single subgroup has been based on the intention of minimizing transfer delay. In other scenarios, it may be appropriate to minimize other parameters. The suitability of a certain metric, 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 files is interested in reducing the financial cost of a transfer. Of course, it could also be suitable to combine several metrics and to weight them according to the intention of the service user. The establishment of Local Groups is also influenced by the structure and the type of a network used for data exchange. At the moment, a new communication service is under development to support application-specific and automated establishment of a group hierarchy.

Related work

Several other approaches have been proposed to provide a reliable multicast service. Early multicast protocols have used sender-based error control to achieve reliability. However, this is not suitable for

large-scale group communication. Recent approaches prefer receiver-based control or hybrid schemes and include special techniques to prevent sender implosion.

Xpress Transport Protocol

The first protocol to incorporate mechanisms for implosion control has been the Xpress Transport Protocol (XTP) [11]. It has defined two heuristics called *damping* and *slotting*. These algorithms suppress redundant control messages by multicasting return messages to the whole group. Hence, every group member receives status messages of other members and skips its own status report if the incoming control unit corresponds to its own state. This mechanism reduces the number of control messages to be processed by the sender. Nevertheless, multicasting control units may be inefficient in large-scale, wide-area networks.

Scalable reliable Multicast

The *Scalable Reliable Multicast (SRM)* [12] enhances damping and slotting mechanisms of XTP to reduce state management overhead. Receivers take solely the responsibility for error correction which is why SRM achieves a high degree of fault tolerance. However, a transmitter is not able to detect the failure of a single receiver. The protocol has been designed for use in the whiteboard tool *wb* [13]. SRM is an example of the receiver-based approach for error control. A receiver missing a certain data unit multicasts a *repair request* to the whole group. Group members that have successfully received the requested packet will multicast it to the entire group. To avoid a flood of *repair requests* and of retransmission, SRM suppress redundant requests by using timers carefully set and adjusted to the current network load. The efficiency of the protocol mainly depends on the correct setting of these timers.

Multicast Transport Protocol

The *Multicast Transport Protocol (MTP)* [14] realizes a centralized control scheme to provide a reliable, totally ordered multicast delivery. Data units from multiple transmitters are delivered in the same order to all group members. A so-called *master* controls data flow by assigning tokens for data transmission. Each potential sender has to obtain a token from the master before transmitting data to the group. This mechanism maintains the global order of data units. Error recovery in MTP is based on negative acknowledgments and retransmissions by the data source.

Reliable Multicast Protocol

The *Reliable Multicast Protocol (RMP)* [15] runs on top of IP multicast and provides a reliable, totally ordered, atomic multicast delivery. It is based on negative acknowledgments that are multicasted to avoid implosion. Reliability is ensured by a rotating token scheme. A single token is passed between group members and designates the site to multicast an acknowledgment for the recently received packets. Missed data units are retransmitted via multicast to all group members.

Other approaches

The *Reliable Multicast Transport Protocol (RMTP)* [16] and the *Tree-based Multicast Transport Protocol (TMTCP)* [17] both use a hierarchical group structure similar to LGC. In contrast to the Local Group Concept, both approaches do not make use of data exchange between neighboring receivers. Instead, both approaches follow strong

hierarchical guidelines and request missed data units always at a higher level controller.

Summary/Conclusion

Common multicast protocols are a significant improvement over simple point-to-point protocols. However, most of them are not suitable for the case where the transmitter has to handle data flow to a large number of receivers. To avoid sender implosion and to increase efficiency of error recovery, the Local Group Concept has been introduced. The benefits are achieved without the necessity to modify internal network equipment such as ATM switches or IP routers. Work is going on to develop a new service for automated establishment of group hierarchies.

Acknowledgments

The author would like to thank J. William Atwood from Concordia University Montreal for valuable discussions. Special thanks to all the members of the LGC Working Group at University of Karlsruhe for valuable comments and suggestions on various subjects of this article.

Authors' address

Markus Hofmann
Institute of Telematics
University of Karlsruhe
Zirkel 2
76128 Karlsruhe
Germany
Fax: +49 721 608 3982
E-Mail: m.hofmann@ieee.org
Web: <http://www.telematik.informatik.uni-karlsruhe.de/~hofmann>

References

- [1] V. Kumar: *Mbone: Interactive Multimedia on the Internet*. New Riders Publishing, Indianapolis, USA, 1995.
- [2] S. Deering: *Host Extensions for IP Multicasting*. RFC-1112, August 1989.
- [3] S. Deering, C. Partridge, D. Waitzman: *Distance Vector Multicast Routing Protocol*. RFC-1075, November 1988.
- [4] J. Moy: *Multicast Extensions to OSPF*. RFC-1584, March 1994.
- [5] C. Huitema: *Routing in the Internet*, Prentice Hall, New Jersey, 1995.
- [6] T. Ballardie, P. Francis, J. Crowcroft: *Core Based Tree (CBT), An Architecture for Scalable Inter-Domain Routing*. ACM-SIGCOM'93, September 1993.
- [7] B. Carpenter: *Architectural Principles of the Internet*; RFC 1958, June, 1996.
- [8] M. Yajnik, J. Kurose, D. Towsley: *Paket Loss Correlation in the Mbone Multicast Network*. UMASS CMPSCI Technical Report # 96-32, University of Massachusetts at Amherst, 1995.
- [9] S. Pejhan, M. Schwartz, D. Anastassiou: *Error Control Using Retransmission Schemes in Multicast Transport Protocols for Real-Time Media*, IEEE/ACM Transactions on Networking, Vol. 4, No. 3, pp. 413-427, June 1996.

- [10] M. Hofmann: A Generic Concept for Large-Scale Multicast. Proceedings of International Zurich Seminar on Digital Communications, Springer Verlag, February 1996.
- [11] W.T. Strayer, ed.: *Xpress Transport Protocol Specification, Revision 4.0*. Available from XTP Forum, Santa Barbara, USA, March 1995.
- [12] 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, August 1995.
- [13] V. Jacobson: *A Portable, Public Domain Network Whiteboard*. Xerox Parc, viewgraphs, April 1992.
- [14] S. Armstrong, A. Freier, K. Marzullo: *Multicast Transport Protocol*. RFC 1301, February 1992.
- [15] B. Whetten, T. Montgomery, S. Kaplan: *A High Performance Totally Ordered Multicast Protocol*. Submitted to INFOCOM'95, April 1995.
- [16] J.C. Lin, S. Paul: *RMTP: A reliable Multicast Transport Protocol*. IEEE INFOCOM'96, 1996.
- [17] R. Yavatkar, J. Griffioen, M. Sudan: *A Reliable Protocol for Interactive Collaborative Applications*. ACM Multimedia'95, 1995.

Markus Hofmann is a research assistant at the Institute of Telematics, University of Karlsruhe. He received his Diploma degree in 1994 from University of Karlsruhe. Currently, he is a member of the High Performance Networking Group and is working on protocol architectures for new generation networks. His research focuses on protocols for multimedia group communication. He can be reached as: m.hofmann@ieee.org