Multicast Transport Protocols: A Survey and Taxonomy

Katia Obraczka *
University of Southern California
Information Sciences Institute
4676 Admiralty Way Suite 1001
Marina Del Rey, CA 90292
voice (310)822-2301 fax (310)823-6714
e-mail katia@isi.edu

November 1, 1997

Abstract

Network support for multicast has triggered the development of group communication applications such as multipoint data dissemination and multi-party conferencing tools. To support these applications, several multicast transport protocols have been proposed and implemented.

Multicast transport protocols have been an area of active research for the past couple of years. This document tries to summarize the activities in this work-in-progress area by surveying several multicast transport protocols. The paper also presents a taxonomy to classify the surveyed protocols according to several distinct features, discusses the rationale behind the protocol's design decisions, and presents some current research issues in multicast protocol design.

1 Introduction

Multicast transport mechanisms have been a topic of intense research and development efforts over the past couple of years. Both the Internet Engineering and Internet Research Task Forces (IETF and IRTF) have been heavily involved in coordinating multicast transport protocol research, development, standardization, deployment, and technology transfer activities. This recent "hype" comes in response to the increasing popularity of groupware applications such as real-time conferencing tools and multipoint data distribution services. Most of the recently proposed multicast transport mechanisms address the requirements of the different types of group communication applications.

Earlier multicast transport mechanisms were designed as a general solution to the group communication problem. More recently it has been recognized that group communication applications have a much wider range of requirements than unicast applications. Therefore a TCP-like solution for the multicast case is not possible: a single, generic protocol simply cannot meet the requirements of all types of multicast applications. This protocol design model was first referred to as Application Layer Framing [8]. We observe that most of the more recently proposed multicast transport mechanisms have been designed with a specific application in mind. For instance, several existing multicast transport protocols address the requirements of delay-sensitive, real-time services, such as multimedia conferencing tools. These applications can tolerate a certain degree of data loss, but are sensitive to packet delay variance. On the other hand, traditional data distribution services, such as multipoint file transfer, are not delay-sensitive, but require that objects be delivered in their entirety or the transfer fails.

The goal of this paper is to present an overview of earlier and current multicast transport protocol research and development. This survey is by no means exhaustive: multicast transport protocols have been an extremely active field and has been in constant evolution for the past couple of years. We review several existing multicast transport

*Katia Obraczka is currently supported by AFOSR award number F49620-96-1-0231, NSF NYI award NCR-9457518, and DARPA grant numbers J-FBI-95-204, DABT63-94-C-0034, and J-FBI-95-155

mechanisms and classify them according to a taxonomy consisting of several distinct features. These features are described in Section 2 below. Section 3 reviews the surveyed protocols including references to commercial product support, and Section 4 summarizes them according to the proposed taxonomy, discusses the rationale behind the protocols' design decisions, and presents some current research issues in multicast protocol design.

Other Sources

A survey and taxonomy that focus on reliable multicast transport protocols can be found in [34]. In [21], five different classes of multicast protocols are identified, namely: sender-initiated, receiver-initiated, receiver-initiated with NACK avoidance, tree-based, and ring-based protocols. This taxonomy is based on how the data buffering mechanism at the source interacts with the error and congestion control mechanisms. Reference lists of multicast protocol related publications can be found in [27, 19].

Reliable multicast frameworks listing important features in reliable multicast protocol design are presented in [12, 38].

2 Taxonomy

We classify multicast transport protocols according to the following features.

- **Data Propagation:** mechanism used to propagate data (not control information) among participating sites.
- **Reliability Mechanism:** how the protocol recovers from data loss.
- **Repair Request:** how the protocol requests retransmission of lost data.
- **Feedback Control:** how the protocol controls the amount of control information receivers generate.
- **Retransmission:** how the protocol propagates data that needs to be retransmitted.
- **Flow and Congestion Control:** mechanism used to prevent the source from overflowing the receivers and from causing network congestion.
- **Locus of Control:** whether the protocol uses a central site to control functions such as message ordering, group membership, and retransmissions.
- **Ordering:** what ordering guarantees the protocol offers and how those guarantees are met.
- **Group Management:** how the protocol manages multicast group membership.
- **Target Application:** whether the protocol was proposed/implemented to support a specific application.

Most multicast transport protocols emphasize scalability as an important design goal. Designing a multicast transport protocol that scales to large group sizes influences the design of several of the functions listed above. For instance, protocols that need to scale should use receiver-based reliability through NACKs and employ some form of feedback control to avoid feedback implosion. Furthermore, protocols targeted at very large groups usually do not assume knowledge of group membership. Sources simply send to a multicast group address and receivers join that group to receive data. Scalability also requires protocol functionality to be distributed to avoid bottlenecks (and of course, single points of failure).

3 Multicast Transport Protocols

Since most of the more recent multicast transport protocols were motivated by specific applications, we present the surveyed protocols grouped by two application classes: multipoint interactive applications and data dissemination applications.

3.1 General Purpose Protocols

Earlier protocols are presented under a "general-purpose" category. RBP, MTP, RMP, and URGC are all message-oriented protocols but were not designed for a specific application.

Reliable Broadcast Protocol

The Reliable Broadcast Protocol (RBP) [7] provides multipoint communication between sites connected by a local-area broadcast network. RBP combines negative and positive acknowledgments to achieve reliability, ordering, and fault tolerance. Messages are multicast to the group through the *token* site. In other words, the token site multicasts an ACK for each message it receives. ACKs inform the sender that the token site received a message. Since ACKs carry a global timestamp, receivers use them to order and detect lost messages.

When a receiver detects a lost message, it sends a NACK to the current token site, which replies with the missing message. For resilience and to limit the amount of state each site needs to keep, the role of the token site rotates among all group members. A member accepts to be the next token site only if it has all the messages earlier than the timestamp of the token passing message. Thus, after a message is generated and the token site is rotated K times, K sites can fail and the message will still be delivered to the sites that are active. Furthermore, the current token site can purge all messages whose timestamps are earlier than the last time this site had the token.

Multicast Transport Protocol

Influenced by RBP, the Multicast Transport Protocol (MTP) [2] provides reliable and globally ordered delivery of client data among a group of communicating processes. MTP is built on top of IP multicasting, and provides strong consistency by multicasting messages under centralized control. One of the participating processes is assigned to be the group master. The group master ensures that data is delivered reliably and in order by giving out transmit tokens and controlling the status of tokens and data messages. It implements an implicit congestion control mechanism by not granting tokens if a certain number of messages are still in the pending state. A message is pending until the master sees all of its packets.

The master is also responsible for supervising group membership. If a group member that is holding a token becomes disconnected, the master removes it from the group and rejects that member's outstanding messages. A process that wants to join the group sends a request to the master, who may accept or reject it. If the request is accepted, the master must be in possession of all the transmit tokens before sending an acceptance message. This way, the master ensures that the new member sees only complete messages. The master may notify its client application about the new member, and it is up to the application to send client state to the new member.

MTP is a NACK-based protocol. When a group member detects a lost message, it sends a NACK to the message producer, who multicasts the requested message to the entire group. However, if the producer has discarded the requested message, it informs the source of the negative acknowledgment. It is up to the application to recover or decide to withdraw from the group. MTP's flow control mechanism is based on a fixed-size transmission window. Every group member agrees on the current window size upon joining a group. This window specifies the maximum number of data packets that a member can send to the group within a specified period of time, and is shared between new and retransmitted packets. Thus, the more retransmissions, the less new packets are injected into the group.

Reliable Multicast Protocol

The Reliable Multicast Protocol (RMP) [39] provides a totally ordered, reliable multipoint transport protocol on top of an unreliable multicast service. Like MTP, RMP evolved from RBP.

Besides using multicast where available, RMP extended RBP with a name service that advertises the existence of multicast groups, and a flow and congestion control mechanism. RMP's name service uses the RMP protocol to manage its distributed database. RMP's group membership protocol allows sites to join or leave a group and updates the members' group membership view accordingly.

RMP's flow and congestion control mechanism uses the algorithms Van Jacobson developed for TCP [17]. Each sender in a group keeps a sliding window regulated by retransmission timers, NACKs from receivers, and positive acknowledgments ACKs from the token site. A sender uses slow-start to increase its window size by 1 packet every time an ACK is received. When a retransmission timer expires signaling congestion, the sender exponentially reduces its window size. To implement flow control, a RMP receiver multicasts a NACK to the group (instead of unicasting it to the token site) when it detects a lost packet. When a sender receives a NACK, it treats it just like an expired timer and backs off exponentially.

RMP is one of the reliable multicast protocols developed and being commercialized by GlobalCast Communications [9]. GlobalCast also developed SRM and RMTP, and provides a common API that interfaces to this family of reliable multicast products.

Xpress Transport Protocol

The Xpress Transport Protocol (XTP) [33] was designed to support a wide range of applications, and thus provide all the functionality of TCP, UDP, and TP4, including services such as transport-level multicast.

XTP provides an explicit multicast group management protocol (MGM) which allows the sending application to control group membership. This includes allowing or denying admission to the group, as well as defining the group's behavior with respect to site failures or data delivery reliability.

Regarding delivery reliability, XTP provides TCP-like fully reliable delivery, UDP-like best-effort delivery, and fast NACKs. In the latter mode, a receiver immediately sends a NACK when it detects out-of-sequence packets, which can improve error repair in environments such as highly reliable networks. In multicast mode, while retransmissions are multicast to the group, NACKs are unicast back to the source, which means that XTP does not provide any feedback control mechanism. XTP allows transmitters to select when receivers should generate acknowledgments, ranging from "always", "sometimes" or "never". If the sender selects to get an ACK from time to time, it will explicitly request the receiver to generate the ACK with its current status information.

XTP offers both window- and rate-based flow control. XTP's rate-based flow control allows a receiver to dynamically pace a transmitter using a rate and burst control parameters, which together establish the inter-packet gap.

Uniform Reliable Group Communication Protocol

The Uniform Reliable Group Communication (URGC) protocol [1] provides reliable and ordered communication among the members of a group. To achieve ordering and decide when to purge old messages, URGC uses centralized control, which rotates among all members of the group. All sites keep a history of all processed messages which allows other sites to recover missing messages. Besides broadcasting its message ordering decisions, the current coordinator also informs the group which is the most up-to-date member. Through unicast communication, other group members can recover missing messages from the most up-to-date site.

Reliable Multicast Frameworks

Recently, the concept of a reliable multicast framework has been introduced. The idea is that instead of a specific protocol targeted at a specific class of reliable multicast applications, a framework for reliable multicast can be used to develop a variety of multicast protocols for applications with different requirements.

Both RMF [12] and RMFP [38] define frameworks for implementing reliable multicast protocols. While RMF introduces the notion of a *universal receiver* which is able to operate with any sender using *self-identifying* packets, RMFP is based on RTP-style profiles. They also differ in how they define their data and control packet formats.

3.2 Support for Multipoint Interactive Applications

Multicast Transport Protocol-2

After their experience implementing and using MTP to support the X window sharing tool Xy, C. Bormann et. al. designed MTP-2 [4], a variant of the original MTP. Their goal was to address the requirements of tools to support remote collaboration, such as teleconferencing services. The modifications introduced by MTP-2 include:

- Immediate joins: unlike MTP, new sites are not required to wait until all transmit tokens are in the master's possession; all messages older than the join message are ignored by a new member.
- Master recovery: MTP-2 addresses the problem of the master being a single point of failure by enabling group members to detect master unreachability and selecting a new group master.
- Dynamic group parameter adjustment: MTP-2 allows the group master to dynamically adjust group parameters such as the heartbeat or the transmission window size. This allows the group to adapt to changes in workload associated with the amount of data generated by the application, and network and server state signaled by increased packet loss or latency.
- Differentiated services: MTP-2 enables source to specify on a per-message basis whether atomic message delivery is needed; if atomicity is not needed, receivers can deliver a message to the application upon arrival of the complete packet sequence.

Real-Time Transport Protocol

The goal of the Real-Time Transport Protocol (RTP) [32] is to support multi-party multimedia conferencing services over the Internet. RTP itself does not guarantee reliable, ordered delivery. It transfers data in a single direction and relies on the Real-Time Control Protocol (RTCP) [32] to support a two-way channel for control information, which includes QoS control.

As a real-time data distribution protocol, RTP tries to avoid packet loss, since once a packet in a real-time data stream is lost, there is not much use in retransmitting it. The control mechanism for real-time transmission proposed in [3] has been implemented as part of the IVS videoconferencing system [36]. It uses the RTP and RTCP suite to carry QoS information. Using the proposed control mechanism, video sources perform probabilistic polling to select a set of receivers that should provide feedback. The selected receivers respond with their measure of the current state of the network. Based on this feedback the video source adjusts its transmission rate. As a side-effect of the probabilistic polling, senders are able to compute an estimated group size, which they inform the receivers. If the size of the group is under a pre-defined threshold, receivers NACK when they detect packet loss. The sender encodes the blocks contained in the lost packet, and sends them to the group.

Scalable Reliable Multicast

Wb, the distributed whiteboard tool presented in [14], uses the Scalable Reliable Multicast (SRM) transport protocol to reliably distribute data among all participating sites. Participants multicast NACKs to request retransmission of lost data, which can be answered by any member that has the information. To avoid generating multiple copies of retransmitted data, retransmissions are multicast to the group. To further reduce the multiple copy problem, a site waits a random period of time before sending a NACK or retransmitting data, and suppresses its own transmission in case it hears it from another member of the group.

A new site joins an ongoing whiteboard conference by announcing its presence with a join message. Current members update their view with the new site. The new site's view of the group is gradually updated through periodic *session* messages generated by current members when the group stays quiet for a certain period of time. Session messages are also used to measure RTT between members, which are used to set the suppression timers.

Log-Based Receiver-Reliable Multicast

The Log-Based Receiver-Reliable Multicast protocol (LBRM) [16] uses a logging server to log all packets transmitted by the source and to retransmit lost packets. The source keeps data that has already been sent until it gets an ACK from the logging server. LBRM uses NACKs between receivers and the logging server and ACKs between the logging server and the source.

Logging servers may be organized in a hierarchy. Secondary logging servers act as a proxy between their clients (local receivers) and the primary logging server. They perform local recovery if they happen to have the packet lost by the local receivers, or request the lost packet from the primary log server on behalf of its clients. Logging servers decide whether they should multicast or unicast a lost packet based on the number of ACKs they get.

Reliable Adaptive Multicast Protocol

The Reliable Adaptive Multicast Protocol (RAMP)'s original design [5] has been extended to operate in an all-optical, circuit-switched, gigabit network environment [20]. RAMP is layered on top of a network-layer multicast protocol, such as IP multicast, and provides NACK-based reliable delivery.

A RAMP sender starts by sending a *connect* message to an IP multicast address. The sender can start multicasting the data after it receives at least one *accept*. Data messages contain monotonically increasing sequence numbers, and once a receiver detects a gap in the message sequence, it requests a retransmission by unicasting a NACK back to the sender. The application selects whether retransmitted messages are multicast to the group or unicast to the requester.

A RAMP sender may operate in 2 modes: burst or idle, and may switch between the 2 modes during the life of a connection depending on the amount of data to be transmitted and the number of receivers. In burst mode, the sender marks the start and end of a burst by setting the ACK flag in the connect message and sending an *idle* message, respectively. By setting the ACK flag, the sender expects reliable receivers [1] to send an ACK back. When the sender does not hear from a known reliable receiver, it retries and eventually ejects the receiver from its membership set. Note that since RAMP assumes a reliable communication medium, it provides a more aggressive recovery mode. In idle mode, the sender multicasts an idle message immediately after it receives the first accept message and when it has no data to send. When a receiver does not receive an idle or data message it assumes data loss and requests retransmission.

RAMP senders manage group membership information, which provides circuit-switched networks with the required information for circuit setup, but does not scale as well as IGMP-style group membership protocols. RAMP uses a simple rate-based flow control in which the sender adjusts its transmit rate based on the feedback received by the slowest receiver in the group.

Transport Protocol for Reliable Multicast

The Transport Protocol for Reliable Multicast (TRM) [31] provides reliable multipoint communication to support interactive multimedia applications such as shared whiteboards and editors. TRM assumes no knowledge of group membership at the source and allows receivers to join and leave an existing multicast group. Group membership management is handled by the application.

A TRM source transmits user data or, when it has no data to send, heartbeat messages containing the sequence number of the last packet sent. Once TRM receivers detect a gap in the packet sequence numbers, they request retransmission by multicasting a NACK to the group. To limit the amount of NACKs generated, a receiver listens for a random period of time and if it sees a NACK requesting the same packet, it suppresses its own. Any group member, including the source, can retransmit the missing data. TRM also uses probabilistic suppression to limit the amount of retransmissions.

[1] A RAMP receiver may choose to operate in *unreliable* mode, in which they ignore message loss.

3.3 Support for Data Dissemination Services

Muse

Muse [22] was developed to multicast news articles on the MBONE [6]. The Muse transmitter accepts a news article through a standard NNTP [18] interaction. Thus for other news feeds, an Muse feed looks just like an NNTP feed. Once a transmitter receives an article and checks it has not seen that article before, it adds a unique identification to the article's header, computes a cryptographic checksum of the article using the RSA MD-5 message digest algorithm, and signs the checksum with its private key. The article is then multicast to the group of Muse receivers as a UDP segment.

When a Muse receiver gets a complete article from the UDP layer (if any packet of the UDP segment is lost, the remaining packets are dropped), it performs the required security checks, and drops the article if any of those checks fail (the article's id and a timestamp are logged to a file). If the article passes the security checks, it is then sent via standard NNTP to the receiving NTTP server, which does not have to be co-located with the site running the Muse receiver.

Muse control messages include a keep-alive message that is multicast to the group if no articles have been sent during a pre-specified period. Muse relies on NNTP or any reliable news propagation protocol for out-of-band recovery of lost articles.

Multicast Dissemination Protocol

The Multicast Dissemination Protocol (MDP) [24] provides a reliable multicast framework for file distribution. MDP evolved from IMM, the image multicaster, used in disseminating satellite image files over the MBone.

The MDP sender fragments the file to be transmitted into a sequence of MDP maximum data units (MDUs) which are multicast to the group using the UDP/IP multicast suite. MDUs are sent at the transmission rate set by the sender application, which can also control the interval between file transmissions. MDP receivers assemble the received data units into the original file, which can be archived or processed using image viewers or text processing tools.

MDP's recovery mechanism works in rounds: after transmitting a file, the source asks receivers for retransmission requests. A receiver who has detected sequence number gaps schedules a retransmission request which gets multicast if no other receiver has requested the same retransmission. MDP also provides an operation mode in which receivers may request repairs at any time during the transmission of a file. The MDP source can also request ACKs from receivers, who respond with information about their current state.

MDP allows sites to join and leave the multicast group. Joining sites announce their presence by multicasting a join packet to the group. Currently, MDP does not have any mechanism to prevent late comers from requesting retransmission of old packets.

Adaptive File Distribution Protocol

Like MDP, the Adaptive File Distribution Protocol (AFDP) [11] provides a reliable file distribution service on top of UDP. AFDP senders, or *publishers* send data to AFDP *subscribers*. A special subscriber in a group, or the group's *secretary*, is responsible for managing group membership, authorizing publishers, and determining the appropriate transmission mode to be used. While AFDP uses multicast as its preferred transmission mode, AFDP packets can also be transmitted using broadcast, or unicast, whenever multicast is not available.

To achieve reliability, AFDP uses NACKs to request retransmission of lost packets. Senders retransmit lost packets to the entire group. AFDP's rate-based flow control mechanism slows down senders every time they receive a NACK. Senders gradually increase their transmitting rate after successful transmissions.

AFDP allows sites to join and leave a distribution group by contacting the group's secretary, but does not provide a recovery mode to handle site failures or network partitions.

Tree-Based Multicast Transport Protocol

The goal of the Tree-Based Multicast Protocol (TMTP) [41] is to provide reliable multicast communication support for 1-to-many bulk data dissemination applications [2]. For error and flow control, TMTP organizes group members into a *control tree*. Each subtree, or *domain* is represented by its root, or the *domain manager*. Domains typically correspond to subnets. At group creation time, the maximum tree degree K, or the maximum number of children at each level, is defined. The control tree grows and shrinks as members join and leave the group. Joining the control tree is based on an expanding ring search to locate the nearest domain manager whose degree is less than K.

A TMTP sender multicasts packets to the corresponding multicast group. The source, as well as the root of each subtree, only waits for ACKs from its children to reclaim buffer space and advance its transmission window. An intermediate node does not have to wait for ACKs from all its children to unicast the ACK to its parent. ACK timeouts trigger a node to retransmit the corresponding packet using limited scope multicast. In addition to this sender-initiated error control approach, TMPTP also relies on NACKs. TMTP receivers use limited-scope NACK multicast subject to suppression.

TMTP uses a window-based flow control scheme and a pre-defined maximum transmission rate set at group creation time.

TMPT supports dynamic group membership, where sites can join and leave a multicast group anytime during the life of a session. A session directory service provides primitives for creating, deleting, joining, and leaving a multicast group. However, knowledge of group membership is not assumed.

Reliable Multicast Transport Protocol

The Reliable Multicast Transport Protocol (RMTP) [23] supports reliable 1-to-many bulk data delivery. Like TMTP, RMTP addresses feedback implosion by organizing group members into a control tree, which governs feedback propagation and processing. Intermediate nodes in the control tree, or *designated receivers (DRs)* are responsible for buffering data received from the source, processing ACKs from their children, and retransmitting lost packets.

Receivers periodically unicast ACKs to their DRs informing them what packets have been correctly received. Depending on how many receivers lost the same packet, the DR decides whether it should multicast or unicast the retransmission.

Like TMTP, RMTP uses window-based flow control and defines a maximum transmission rate set at group establishment time. RMTP allows dynamic group membership: receivers may join and leave a multicast group any time during a session. However, knowledge of group membership is not required. RMTP not only allows receivers to join at any time, but also guarantees that they receive the entire data reliably. This feature comes at the price of having the DRs buffer transmitted data during the whole session.

RMTP congestion control mechanism is based on shutting down the transmission window to 1 packet if the number of ACKs with retransmission requests exceeds a pre-defined threshold.

Lucent Technologies' e-cast [35] reliable multicast product family is based on RMTP.

The Multicast File Transfer Protocol

StarBurst's Multicast File Transfer Protocol (MFTP) [26] is another multicast protocol that supports reliable 1-to-many data transmission. The MFTP source may operate in unicast, multicast or broadcast mode depending on what kind of information is being sent and what transmission modes are supported by the network. The MFTP client is able to handle any transmission mode.

MFTP consists of a group management and data transmission protocols. Using the data transmission protocol, the MFTP sender announces a file transfer session periodically during the announcement phase. In response to announcements, clients register to receive the file MFTP sends data in passes: in the first pass, it sends the entire

[2] In [41], the authors refer to dissemination services in the context of interactive applications.

file at a pre-defined transmission rate. In subsequent passes, the server retransmits lost packets. At the end of each pass, the source requests status from receivers who then respond in the form of ACKs in case they have received all packets , or NACKs if packets have been lost. Receivers only send ACKs if explicitly required by the source, otherwise, they only send NACKs if they need packets to be retransmitted. The source may specify an address different than its own where receivers should send ACKs and NACKs. This is a way of aggregating receiver feedback to avoid implosion at the source. Feedback suppression is optional and is only used with NACKs: receivers wait for a random interval between 0 and n before multicasting its NACK to the specified response address. The value n is a configurable protocol parameter.

MFTP's group management protocol supports a *public group* to which announcements are sent. Data is transmitted on the private group address. While any client may join and listen to announcements, only clients authorized by the source may join the private group. MFTP uses two group management models: open and closed. In closed groups, the source knows which clients are authorized to receive the transmission at announcement time. In the open group model, the source does not know a priori who is allowed to receive the data. Any clients may request to participate in data delivery. The source may limit receiver participation (limited open group), or allow any client to participate (unlimited open group). In the closed and limited open group cases, group membership is known by the source. This is not the case in the unlimited open group model.

MFTP is a product developed by StarBurst Communications [10] and supports StarBurst's family of multipoint file transfer software.

3.4 Interactivity versus Reliability

Interactive and dissemination services differ in their delivery delay (or delay variance) and reliability requirements. While interactive applications are willing to sacrifice reliability in favor of real-time delivery, data dissemination tools tolerate higher delivery delays but require reliable delivery.

Recently the *resilient* multicast delivery model has been proposed to address the different reliability and interactivity requirements among different applications, as well as among different receivers in the same multicast group. The idea is that in applications such as interactive teleconferencing, different participants have different levels of participation: ones are purely spectators, while others are more active. For instance, in IETF broadcasts, there are usually many more passive participants than active ones. Passive receivers would rather have high playback quality which can be achieved by delayed playback combined with retransmissions.

Structure-Oriented Resilient Multicast

Like TMTP and RMTP, the Structured-Oriented Resilient Multicast (STORM) [40] protocol performs error recovery by organizing group members into a dynamic distribution graph (not necessarily a tree). A group member keeps a list of parent nodes and as soon as it notices a packet loss, it unicasts a NACK to each parent at a time. If it does not hear from that parent within a given interval, it tries the next parent until it gets the packet or the packet becomes obsolete. Upon receiving a NACK from a child, a node checks whether it has the required packet, in which case it unicasts it to the requestor.

STORM's error recovery graph changes with changes in group membership. When a node joins the multicast group, it performs an extended ring search (ERS) using limited scope multicast to look for parents. When a group member receives a ERS message, it unicasts back a reply containing a measure of its perceived loss rate as a function of its playback buffer size. The larger the playback buffer, the smaller the loss rate, since a larger buffer allows delayed and retransmitted packets to arrive in time to be palyed back. A new member choose their parents based on its own playback buffer size and the parent's perceived loss rate.

To adapt to changes in loss rates, nodes periodically measure its parents' performance by computing the number of times a parent answered retransmission requests. If a parent's performance drops below a threshold, the parent is removed from the parent's list. A replacement is found by performing another ERS, or by choosing a node that may have replied to a previous ERS but not yet used as parent. STORM also provides a mechanism to avoid loops in its error recovery structure.

Protocol	Data Propagation	Reliability Mechanism	Repair Request	Feedback Control	Retrans-mission	Flow Control	Locus of Control	Ordering	Group Management	Target Application
RBP	B-cast	ACKs-NACKs	U-cast	-	U-cast	-	Central'd	Total	Explicit	Section 4.9
MTP MTP-2	M-cast	NACKs	U-cast	-	M-cast	Rate	Central'd	Total	Explicit	Section 4.9
RMP	M-cast	ACKs-NACKs	M-cast	-	U-cast	Window	Central'd	Total	Explicit	see Section 4.9
XTP	U-cast-M-cast	ACKs-NACKs	U-cast	-	U-cast-M-cast	Rate	Central'd	Section 4.7	Explicit	Section 4.9
URGC	B-cast	NACKs	U-cast	-	U-cast	-	Central'd	Total	Explicit	Section 4.9
RTP	M-cast	Section 4.2	Section 4.2	Probabilistic polling	Section 4.2	Rate	Distrib'd	-	-	Interactive Multimedia Applications
SRM	M-cast	NACKs	M-cast	Timer-based	M-cast	-	Distrib'd	-	Explicit	Interactive Multimedia Applications
LBRM	M-cast	ACKs-NACKs	U-cast	Structure-based	U- or M-cast	-	Central'd	-	Implicit	Interactive Multimedia Applications (Distributed Interactive Simulations)
RAMP	M-cast	NACKs	U-cast	-	U- or M-cast	Rate	Distrib'd	Section 4.7	Explicit	Interactive Multimedia Applications (Optical Gigabit Network)
TRM	M-cast	NACKs	M-cast	Timer-based	M-cast	Window	Distrib'd	Section 4.7	Implicit	Interactive Multimedia Applications
Muse	M-cast	NACKs	U-cast	-	U-cast	-	Distrib'd	-	Implicit	Netnews article propagation
MDP	M-cast	NACKs	M-cast	Timer-based	M-cast	Rate	Distrib'd	Section 4.7	Implicit	Data (File) Distribution
AFDP	M-cast	NACKs	U-cast	-	M-cast	Rate	Central'd	Section 4.7	Explicit	Data (File) Distribution
TMTP	M-cast	ACKs-NACKs	U-cast	Structure-based	M-cast (limited)	Window	Distrib'd	-	Implicit	Data Dissemination
RMTP	M-cast	ACKs	U-cast	Structure-based	U- or or M-cast	Window	Distrib'd	-	Implicit	Data Dissemination
MFTP	U-, M-, or B-cast	ACKs-NACKs	U-cast(ACKS) M-cast(NACKs)	Timer-based	M-cast	Window	Distrib'd	-	Explicit	Data Dissemination
STORM	M-cast	NACKs	U-cast	Structure-based	U-cast	Window	Distrib'd	-	Implicit	Continuous Media

Table 1: A taxonomy of multicast transport protocols.

4 Classification

Table 1 summarizes the surveyed protocols according to the characteristics described in Section 2. These features establish several dimensions in the multicast transport protocol design space. We discuss them in light of the surveyed protocols and examine the rationale behind the protocols' design decisions. We also present some current multicast protocol research issues.

4.1 Data Propagation

Most of the surveyed protocols use IP multicast as their preferred data propagation mechanism, but will work with regular unicast when IP multicast is not supported.

4.2 Reliability Mechanism

Traditional reliable unicast transport protocols, such as TCP, use positive acknowledgments (ACKs) to recover from packet loss. This approach to achieving reliability is often referred to as sender-initiated, since it is the responsibility of the sender to detect packet losses. In a multicast environment, as group sizes increase, the sender-initiated scheme may cause ACK implosion since each delivered packet triggers an acknowledgment from every receiver in the group.

Alternatively, in the receiver-initiated approach to reliability, receivers detect packet losses and request its retransmission by generating a negative acknowledgment (NACK). Placing the responsibility of recovering from packet losses on the receiver helps alleviate the ACK implosion problem. The performance comparison study presented in [29] confirms that receiver-initiated multicast transport protocols deliver better performance than their sender-initiated counterparts.

Most reliable multipoint transport protocols are either pure receiver-initiated or use a hybrid approach by combining receiver- and sender-initiated reliability.

Except for RTP, all surveyed protocols offered reliable delivery. RBP, RMP, LBRM, TMTP, and MFTP achieve reliability using both ACKs and NACKs, while the other protocols use NACKs only. To avoid feedback implosion at the source, these protocols designate a special site which among other functions is responsible for acknowledging the receipt of every message sent by the source and responding to retransmission requests. The other receivers in the group use NACKs to request retransmission of lost messages.

RTP does not guarantee reliable delivery. Instead, it tries to avoid losses by adjusting the sender's rate based on feedback from the receivers. However, in the case where the number of receivers is below a pre-defined threshold, the feedback control mechanism allows receivers to NACK when they detect packet loss.

There has been recent work on using Forward-Error Correction (FEC) techniques to address reliability in multicast delivery. The idea is to anticipate that there will be losses and transmit redundant data. Receivers can then reconstruct the original data in the event of a certain amount of packet loss. Clearly, there is a tradeoff between the overhead in processing (both at the sender and at the receiver) and transmitting redundant data and the ability to recover from losses without having to request retransmissions and retransmit lost packets. Both [28, 30] evaluate the performance of FEC-based error recovery mechanisms.

4.3 Repair Request and Retransmission

In reliable transport protocols, repair requests and retransmissions may be propagated point-to-point or to the whole group. In protocols such as MTP and AFDP, a receiver unicasts a NACK to the sender to request retransmission of a lost packet. In response, the sender multicasts the repair to the whole group assuming that other receivers may have lost the same packet. Other protocols like SRM, go a step further, and use multicast to propagate NACKs as well. The idea is to suppress NACKs from other receivers that lost the same packet.

Protocols like LBRM, SRM and TMTP propose the concept of *local recovery*. The idea is to try to recover from packet loss without having to go all the way to the source. In LBRM, logging servers are responsible for packet retransmissions may be organized hierarchically to recover the set of receivers for which they proxy. In SRM, *local loss neighborhoods* are not as clear cut and are the subject of ongoing research. TMTP uses limited scope multicast to delimit the scope of multicast retransmissions.

4.4 Feedback Control

Receiver-based reliable multicast protocols can still cause feedback implosion, especially when losses occur higher up in the multicast tree in larger groups over lossy networks.

To avoid feedback implosion, multicast protocols should provide some form of feedback control, i.e., a mechanism that restricts the amount of feedback generated by the multicast group.

In [15], solutions to the feedback implosion problem are classified as *structure-based* or *timer-based*. Another classification based on how to limit the amount of feedback generated by group members is proposed in [21]. In the first classification, structure-based approaches rely on a designated site (either a dedicated server in the case of the LBRM Protocol or a pre-assigned group member) to process and filter feedback information. The second classification provides a more general definition of structure-based feedback control. It refers to solutions such as the ones adopted by TMTP and RMTP, which organize multicast group members into some structure in order to filter the amount of feedback generated by the group

Timer-based solutions rely on probabilistic feedback suppression to avoid implosion at the source. Receivers in the SRM protocol delay their retransmission requests for a random interval, uniformly distributed between the current time and the one-way trip time to the source. The goal is that group members closer to the source send their feedback sooner suppressing feedback from farther away members. A site uses periodic *session messages* to measure its distance (based on the resulting RTT) to the other group members.

The feedback control mechanism used by the IVS videoconferencing tool and layered atop of RTP does not fall into either the structure-based or the timer-based categories. In this approach to feedback control, video sources use probabilistic polling to select a set of receivers that should provide feedback.

4.5 Flow and Congestion Control

Several of the surveyed protocols do not implement any explicit form of flow/congestion control mechanisms. They rely on the fact that the retransmission flow will grab part of the available bandwidth and cause the data flow to slow down. MTP implements a more explicit flow control mechanism by not granting transmit tokens if a certain number of messages are outstanding. RMP uses a TCP-like window-based flow control mechanism, while RTP, AFDP, and RAMP implement rate-based flow control schemes.

Flow and congestion control for multicast protocols is a hard problem and an active research area. Since providing congestion control mechanisms is critical in enabling reliable multicast transport protocols to be deployed on the Internet, it has been considered a priority area by the IRTF Reliable Multicast working group. Only a few few existing multicast transport protocols implement explicit congestion control. Layered Multicast [25] addresses the congestion problem in the context of video stream multicast transmission, an example application that can tolerate data loss. Streams are decomposed into several layers. Encoding is done such that the composition of layers leads to higher quality video. The lowest layer in the stream is a low quality, low bandwidth stream. Each successive layer provides more information that when combined with the lower layers results in a higher quality image. The more bandwidth that is available, the more layers one can receive. As congestion is detected at the receiver via packet loss, the receiver unsubscribes to higher bandwidth groups. Periodically, receivers that are not experiencing congestion, join higher bandwidth groups to take advantage of available network bandwidth.

The congestion control mechanism for bulk-data transfer applications presented in [37] uses forward error correction (FEC) techniques to stripe the data into multiple, cumulative, redundant layers. Each layer is sent to a different multicast address. Receivers choose their receiving rate by joining/leaving the appropriate multicast groups. The more layers receivers subscribe, the faster they receive the data.

Our representative-based congestion control mechanism [13] uses a small, dynamic set of group members, or *representatives*, to provide timely feedback on behalf of congested subtrees of a multicast distribution tree. Our algorithm is based on the assumption that in a multicast group, a small set of bottleneck links will cause the majority of the congestion problems. We dynamically select a small set of group members to represent the congested multicast subtrees. This small set of representatives allows us to apply the point-to-point congestion control model to a multicast group of arbitrary size. These group representatives provide immediate feedback which suppresses feedback from other receivers, thus preventing feedback implosion at the source. In addition to suppression, representative feedback allows the source to concentrate its congestion control efforts on the congested subtrees. Based on the feedback from representatives, the source adjusts the current transmission rate. As new congestion appears in the tree, new representatives are selected and old ones dropped from the representative set.

4.6 Control

Several of the surveyed protocols assign certain control functions to a central site. Both RBP and RMP use a central site to avoid ACK implosion at the source and to provide message ordering to the whole group. Since MTP is a NACK-only protocol (it does not funnel ACKs through a central site), it uses a central site to control message ordering and manage group membership. AFDP's secretary site manages group membership and controls the number of active senders. In the case of LBRM, logging servers log all packets sent by the source and retransmit lost packets.

4.7 Ordering

Message-oriented protocols such as RBP, MTP, RMP, and URGC guarantee totally ordered delivery by having a central site assign global message sequence numbers.

Muse relies on NNTP for ordering while AFDP senders assign monotonically increasing sequence numbers to the packet stream corresponding to the file being propagated. AFDP receivers use the sequence numbers to order the packets received and to detect lost packets. XTP, MDP, RAMP, and TRM also use sequence numbers to order transmitted segments.

4.8 Group Management

Some multicast protocols do not require knowledge of group membership: sources simply send data to a multicast group address to which receivers can join to receive the data. This implicit group management model matches the way IP multicast handles group membership and has better scalability properties than explicit group management. Some protocols like RTP provide no group membership management, leaving it up to the application.

Other protocols may require full control over group membership. In this case, membership is either static throughout a session, or an explicit group membership protocol ensures reliable joins and leaves. A variation of the latter approach is to have sites reliably join a group but letting them leave silently. An example application that can use this group management approach is data dissemination with access control requirements. In these scenarios, the source needs to authorize new members but does not have to be aware of members leaving the group. StarBurst's MFTP implements this kind of explicit group management.

4.9 Target Application

While earlier multicast transport mechanisms such as RBP, MTP, URGC, and XTP were proposed as a general solution to the group communication problem, more recent protocols have been designed with a specific application in mind.

For instance, protocols like SRM address the requirements of delay-sensitive, real-time services, such as multimedia conferencing tools. These applications can tolerate a certain degree of data loss, but are sensitive to packet delay variance. LBRM's design goals target a slightly different real-time application: distributed interactive simulation, which normally does not generate frequent updates, yet the time to propagate updates to all participants is critical.

On the other hand, protocols like AFDP, MDP, Muse, TMTP, RMTP, and MFTP target services that are not delay-sensitive but require that objects be delivered in their entirety, or the transfer fails.

STORM introduces the idea of resilient, as opposed to reliable, multicast. It allows different receivers in the same multicast group to establish their own reliability requirements. It targets continuous media applications.

5 Summary and Conclusions

This paper surveys the ever evolving area of multicast transport protocols. It reviews several existing multicast protocols, and proposes a classification scheme based on several dimensions in the multicast transport protocol design space. This classification includes features such as data propagation, reliability, ordering, flow, congestion and feedback control, group management, and their target application. In light of the proposed classification scheme, the paper discusses the design decisions taken by the surveyed protocols and presents current research issues in multicast protocol design.

From the several surveyed protocols, we observe the application-level framing trend in multicast protocol design: more recent protocols target the requirements of specific, constantly evolving, group communication applications. Constantly evolving applications are one of the reasons why there are still many issues that need to be resolved before reliable multicast transport are considered Internet-safe. Both the Internet Engineering and Internet Research Task Forces (IETF and IRTF) have been heavily involved in coordinating multicast transport protocol research, development, standardization, deployment, and technology transfer activities.

Because their one-to-many or many-to-many delivery model, reliable multicast protocols can have considerable negative performance impact on the underlying network. They can generate large amounts of data as well as control traffic, which may cause network congestion and eventually lead to congestion collapse. Furthermore, multicast protocols may try to inject packets at a rate, which the network, given its current state, cannot tolerate. These negative performance effects become more acute as multicast groups size grows. Therefore, effective congestion control mechanisms are one of the priorities in the reliable multicast research community's agenda. Scalable error recovery techniques as well as security mechanisms are other current research issues.

References

[1] R. Aiello, E. Pagani, and G.P. Rossi. Design of a reliable multicast protocol. *Proc. of the IEEE Infocom'93*, pages 75–81, March 1993.

[2] S. Armstrong, A. Freier, and K. Marzullo. Multicast transport protocol. Internet Request for Comments RFC 1301, February 1992.

[3] J.C. Bolot, T. Turletti, and I. Wakeman. Scalable feedback control for multicast video distribution in the internet. *Proc. of the Conference on Communications Architectures, Protocols, and Applications, ACM SIGCOMM 1994*, August 1994.

[4] C. Bormann, J. Ott, H.-C. Gehrcke, T. Kerschat, and N. Seifert. MTP-2: Towards achieving the S.E.R.O. properties for multicast transport. *Proceedings of the International Conference on Computer Communications and Networks (ICCCN) 94*, September 1994.

[5] R. Braudes and S. Zabele. Requirements for multicast transport protocols. Internet Request for Comments RFC 1458, May 1993.

[6] S. Casner. Frequently asked questions (FAQ) on the multicast backbone (MBONE). On-line documentation; available from ftp://venera.isi.edu/mbone/faq.txt, January 1993.

[7] J.M. Chang and N. F. Maxemchuk. Reliable broadcast protocols. *ACM Transactions on Computer Systems*, 2(3):251–273, August 1984.

[8] D. Clark and D. Tennenhouse. Architectural considerations for a new generation of protocols. *Proc. of the Conference on Communications Architectures, Protocols, and Applications, ACM SIGCOMM 1990*, September 1990.

[9] GlobalCast Communications. Globalcast communications reliable multicast protocols. http://www.gcast.com/reliablemulticast.html.

[10] StarBurst Communications. Starburst multicast. http://www.starburstcom.com/sbmcast.htm.

[11] J.R. Cooperstock and S. Kotsopoulos. Why use a fishing line when you have a net? an adaptive multicast data distribution protocol. *In Proceedings of Usenix '96*, 1996.

[12] B. DeCleene, S. Bhattacharaya, T. Friedman, M. Keaton, J. Kurose, D. Rubenstein, and D. Towsley. Reliable multicast framework (rmf): A white paper. March 1997.

[13] Dante DeLucia and Katia Obraczka. Multicast feedback suppression using representatives. *Proceedings of the IEEE Infocom'97*, April 1997.

[14] S. Floyd, V. Jacobson, S. McCanne, C.G. Liu, and L. Zhang. A reliable multicast framework for light-weight sessions and application-level framing. *1995 ACM Sigcomm Conference*, pages 342–356, October 1995.

[15] M. Grossglauser. Optimal deterministic timeouts for reliable scalable multicast. *Proc. of the IEEE Infocom'96, San Francisco, CA*, pages 1425–1432, March 1996.

[16] H.W. Holbrook, S.K. Singhal, and D.R. Cheriton. Log-based receiver-reliable multicast for distributed interactive simulation. *1995 ACM Sigcomm Conference*, pages 328–341, October 1995.

[17] Van Jacobson. Congestion avoidance and control. *ACM SIGCOMM 88*, pages 273–288, 1988.

[18] B. Kantor and P. Lapsley. Network news transfer protocol - a proposed standard for the stream-based transmission of news. Internet Request for Comments RFC 977, February 1986.

[19] J.P. Knoght. Multicast transport protocols. http://hill.lut.ac.uk/DS-Archive/MTP.html, 1997.

[20] A. Koifman and S. Zabele. Ramp: A reliable adaptive multicast protocol. *Proc. of the IEEE Infocom'96, San Francisco, CA*, pages 1442–1451, March 1996.

[21] B.N Levine and J.J. Garcia-Luna-Acceves. A comparison of known classes of reliable multicast protocols. *Proceedings of the International Conference on Network Protocols (ICNP)'96*, October 1996.

[22] K. Lidl, J. Osborne, and J. Malcolm. Drinking from the firehose: Multicast USENET news. *Proceedings of the 1994 Winter USENIX Conference*, 1994.

[23] J.C. Lin and S. Paul. Rmtp: A reliable multicast transport protocol. *Proceedings of the IEEE INFOCOM'96*, pages 1414–1424, March 1996.

[24] J. Macker and W. Dang. The multicast dissemination protocol (mdp) framework. Internet Draft, Internet Engineering Task Force, draft-macker-mdp-framework-00.txt, November 1996.

[25] S. McCanne and V. Jacobson. Receiver-driven layered multicast. *1996 ACM Sigcomm Conference*, pages 117–130, August 1996.

[26] K. Miller, K. Robertson, A. Tweedly, and M. White. Starburst multicast file transfer protocol (mftp) specification. Internet Draft, Internet Engineering Task Force, draft-miller-mftp-spec-02.txt, January 1997.

[27] T. Montgomery. Reliable multicast links. http://research.ivv.nasa.gov/RMP/links.html, October 1997.

[28] J. Nonnenmacher, E. Biersack, and D. Towsley. Parity-based loss recovery for reliable multicast transmissions. *Proceedings of the ACM Sigcomm'97*, pages 289–300, September 1997.

[29] S. Pingali, D. Towsley, and J. F. Kurose. A comparison of sender-initiated and receiver-initiated reliable multicast protocols. *1994 ACM SIGMETRICS Conference*, May 1994.

[30] L. Rizzo and L. Vicisano. A reliable multicast data distribution protocol based on software fec techniques. *Proceedings of the High Performance Communication Systems (HPCS) '97*, June 1997.

[31] B. Sabata, M.J. Brown, and B.A. Denny. Transport protocol for reliable multicast: TRM. *Proceedings of the IASTED International Conference on Networks*, pages 143–145, January 1996.

[32] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. Internet Request for Comments RFC 1889, January 1996.

[33] T. Strayer. Xtp web page. http://www.ca.sandia.gov/xtp/xtp.html.

[34] TASC. Reliable multicast protocols. http://www.tascnets.com/mist/doc/mcpCompare.html, August 1996.

[35] Lucent Technologies. e-cast reliable multicast software. http://www.bell-labs.com/project/e-cast/.

[36] T. Turletti. H.261 software codec for video conferencing over the internet. INRIA Research Report 1834, January 1993.

[37] L. Vicisano and J. Crowcroft. One to many reliable bulk-data transfer in the mbone. June 1997.

[38] Z. Whang, J. Crowcroft, C. Diot, and A. Ghosh. Framework for reliable multicast application design. *Proceedings of the HIPPARCH'97*, 1997.

[39] B. Whetten, T. Montgomery, and S. Kaplan. A high performance totally ordered multicast protocol. *Theory and Practice in Distributed Systems, Springer Verlag, LCNS 938.*

[40] X.R. Xu, A.C. Myers, H. Zhang, and R. Yavatkar. Resilient multicast support for continuous-media applications. *Proceedings of the NOSSDAV1997*, 1997.

[41] R. Yavatkar, J. Griffioen, and M. Sudan. A reliable dissemination protocol for interactive collaborative applications. *Proceedings of the ACM Multimedia'95*, 1995.