

# End-to-End Reliable Multicast Transport Protocol Adaptation for Floor Control and Other Conference Control Functions Requirements

*Nadia Kausar, Jon Crowcroft*  
*Department of Computer Science*  
*University College London*  
*Gower Street, London WC1E 6BT, UK*  
*Phone +44171 504 4433*  
*Fax +44171 387 1397*  
*n.kausar@cs.ucl.ac.uk, jon@cs.ucl.ac.uk*

## Abstract

In order to provide guaranteed QoS multiparty collaborative multimedia applications require reliable transmission of data. The multimedia applications can vary from distributed games, shared whiteboard to interactive video conferencing. These applications often involve a large number of participants and are interactive in nature with participants dynamically joining and leaving the applications[Sudan95]. In order to provide many-to-many interaction when the number of participants is large IP multicast is a very good option for communication. IP multicast provides scalability and efficient routing but does not provide the reliability these multimedia applications may require. Though a lot of research has been done on reliable multicast transport protocol, it really seems that the only way of doing a reliable multicast is to build it for a given purpose like conference control in multimedia conferencing.

This paper compares some of the available multicast transport protocols and analyses the most suitable features and functionalities provided by these protocols for a facet of conference control, floor control. The goal is to find or design a reliable multicast transport protocol which would scale to tens or hundreds of participants scattered across the Internet and deliver the control messages reliably.

## Keywords

**Floor control, reliable multicast, conference control**

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35388-3\\_42](https://doi.org/10.1007/978-0-387-35388-3_42)

H. R. van As (ed.), *High Performance Networking*

© IFIP International Federation for Information Processing 1998

## 1 INTRODUCTION

Conferences come in many shapes and sizes, but there are two models of conference control. These are known as Formal/tightly coupled conferencing and Informal/loosely coupled conferencing. Lightweight/informal sessions are multicast based multimedia conferences that lack explicit conference membership control and explicit conference control mechanisms. Typically a loosely coupled session consists of a number of many-to-many media streams supported using RTP and RTCP using IP multicast. Typically, the only conference control information that is provided during the course of a light-weight session is that distributed in the RTCP session information, i.e. an approximate membership list with some attributes per member.

On the contrary, tightly coupled conferences where the media streams are flowing from mainly one-to-many or one-to-one basis, requires an explicit conference control mechanism. In a model like that a user interface is provided where the chair can choose to give a floor to one of the participants, so one person can talk, take control of the shared whiteboard or use the video channel at a time.

The most conventional tightly coupled conferences are ITU based H.323[H.323] or T.120[T.120] standard conferencing which was initially designed to work over circuit switched networks like ISDN and the loosely coupled conferences are Mbone[MboneFAQ] based which are designed for IP multicast. Some features of the tightly coupled conferences like floor control have only recently been designed to work on IP with TCP over it or use UDP for other type of data. Therefore, the most suitable reliable IP multicast for tightly coupled conferences is a recent issue.

IP Multicast provides a service model by which a group of senders and receivers can exchange data without the senders needing to know who the receivers are\*, or the receivers needing to know in advance who the senders are. Hosts that have joined a multicast group will receive packets sent to that group. Therefore, this service model can lead to applications which will scale to hundreds/thousands or more receivers. Although, because of the limited bandwidth most applications like videoconferencing will deploy floor control to limit traffic from the group to a small number of concurrent sources.

In order to support floor control either for a tightly coupled session (where reliability and ordering of the messages may get the highest priorities) or a loosely coupled session (where congestion control or retransmission strategy may be more complex and more critical than strict ordering), certain characteristics from a multicast protocol are required. The requirements for conference control from a transport protocol are:

1. Reliability and loss detection
2. Retransmission strategy, queue management
3. Scalability - source to many receivers, many sources to many receivers etc

\* Unless a higher level agreement has been done.

### Ordering

4. Scope of membership
5. Congestion control
6. Integrated security

A lot of research is being done on reliable multicast transport protocols. This paper looks at some of the available protocols like SRM, MTP/SO, RMTP, RLC and PGM and compares them against the requirements of single facet of conference control Floor Control. The reason for choosing these particular protocol is that they provide a lot of the functionalities required by a conference control mechanism. However, there may well be other protocols available now or may well be in the design phase which may serve the same purposes.

Section 2.0 is the background of some of the available reliable multicast protocols, section 3 analyses floor control and its requirements in general, the following section looks at the limitation of floor control, section 6.0 highlights the limitations of some of these multicast transport protocols in the light of floor control requirements. The last sections (7.0 and 8.0) describes an ideal reliable IP multicast with certain characteristics and which of the available protocols provide some of these functionalities.

## 2 BACKGROUND

Loss detection and retransmission strategy are two important aspects in the design of any reliable protocol. In a reliable transport protocol a recipient can (within bounded time) find out when it is failing or being partitioned from active senders. A sender is assured (with sufficient probability) that all its messages reach within bounded time.

In a traditional point-to-point reliable protocol such as TCP, positive acknowledgements are used to detect loss and the sender is responsible for retransmission of the packet. Using TCP one can provide HTTP Web traffic, FTP file transfers, and e-mail. All TCP traffic is unicast, that is it has one source and one destination. The nature of data can be either bulk data transfer where all data is sent one way and then the sender waits for a response or interactive where as soon as each data unit is sent acknowledgement has to be returned. The transmitter sends out a window's worth of data before requiring an acknowledgement.

It is harder to transfer data "reliably" from source(s) to R receivers (where R can be 10's to 100,000 or more), because multicast protocols interact with multiple parties simultaneously and so involve a higher number of links. Therefore, the likelihood is greater that some of the paths in the source's multicast tree are unstable at any time. In addition, the instability in any portion of the multicast tree may affect many members of the group because of the collaborative adaptive algorithms used[Floyd98]. In particular, it is difficult to build a generic reliable transport protocol for multicast, much as TCP is a generic transport protocol for Unicast. Reliable multicast is a case where "one size fits all" does not work at all.

Applications often have very different reliability and latency requirements, state management styles, error recovery and group management mechanisms. A reliable multicast transport protocol that meets the worst-case requirements is unlikely to be efficient and scalable for many application requirements[Zhang97].

In a teleconferencing environment, a desirable robustness property is the ability to continue operating within partitions should the group become partitioned. Ultimately, the applications that use the multicast transport platform should be the ones to decide when the situation has deteriorated to a point where continuing is meaningless.

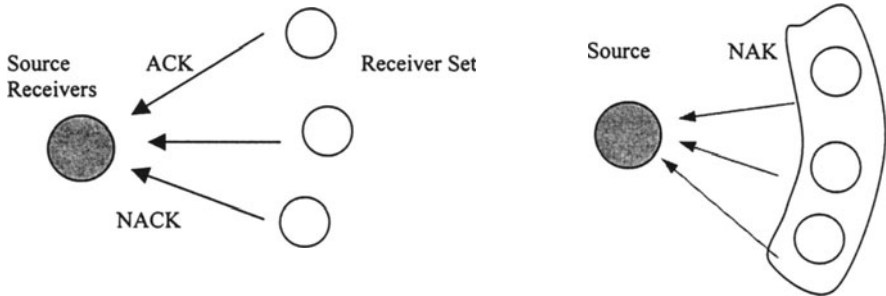


Diagram 1: A basic diagram of a sender initiated Protocol

Diagram2: Receiver initiated Protocol

The design of a reliable IP multicast can be based on either a tree-based, a ring based or an ACK/NACK i.e. acknowledgement structure.

In the following subsection we provide an overview for some of the reliable multicast transport protocols:

## 2.1 SRM

Scalable reliable multicast (SRM) has been embedded into an Internet collaborative whiteboard application called wb. In SRM, whenever a receiver detects a packet loss, it multicasts a NACK packet to the entire group. Upon receiving the NACK packet, any member holding the desired packet can multicast it to the group. To avoid duplicate NACK and repair packets, a suppression algorithm is used in which a node sets a random timer before multicasting a NACK or repair packet. The messages specify a time-stamp used by the receivers to estimate the delay from the source, and the highest sequence number generated by the node as a source. SRM's implementation requires that every node stores all packets or that the application layer stores all relevant data.

One of the problems with SRM is that this algorithm will end up consuming a lot of bandwidth when there is little correlation of losses among receivers. For example, in a group of 1000 receivers, when only one receiver loses a packet, all 1000 receivers need to process the multicast NACK and repair packets. This

causes significant overhead. Also if one set of hosts in particular requires a packet, it is not desirable to multicast the packet to all the possible groups. One possible method of improving SRM's efficiency is to use localised recovery. The idea is to multicast NACKs and repairs locally to a limited area instead of to the whole group. Using the TTL (Time to Live field in the IP packet header is one possible way to implement scope control.

## 2.2 MTP/SO

Multicast Transport protocol or MTP provides an atomic and reliable transmission of messages. MTP/SO provides global ordering where messages are assigned to different streams. Therefore the delay caused by global ordering (for example when a short message is preceded by a very long one) is eliminated. MTP/SO proposes self-organisation of the members of a group into local regions for addressing the NACK implosion problem. MTP/SO provides a rate controlled transmission of user data. There are three main groups of members within a group: co-ordinator, repeaters and normal members. To provide maximum throughput the co-ordinator can send and receive retransmission, whereas if it is a type of a member who is just 'listen only' capable, the only packet type they can send to the group is unreliable multicast datagrams.

The rate controlled transmission of user data is very useful for floor control. If only few users are capable of holding the floor then there is only little point of giving all the other 10,000 receivers the capability of asking for retransmission of floor request. Although a lot of the functionalities of this protocol can be used for conference control (which is discussed in the section) purposes, the implementation of MTP/SO is in very early stage yet.

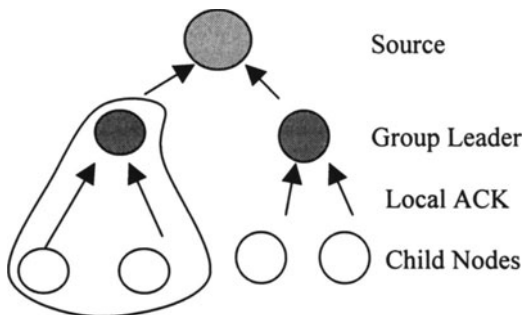


Diagram 3: A basic diagram of a tree based protocol

### 2.3 RMTP (Globalcast Communication)

Reliable Multicast Transport Protocol (RMTP) organises all the nodes into a tree structure. The receiving nodes are always at the bottom of the tree. Ideally the senders are at the top. The sender transmits messages using IP multicast, after a message is transmitted the sender will not release the memory until it receives a positive acknowledgement from the group. The receivers do not send acknowledgement directly to the top node(sender), but send hierarchical acknowledgements (HACKs). A receiver transmits a HACK to their parent in the tree structure. The parent gathers all HACKs from its children and sends a HACK to its parent node one step higher in the tree. The HACKs are propagated upward to the top of the tree and the sender is eventually notified. This design allows dissemination of messages to a large number of receivers without causing ACK implosion.

If there are lots of listeners and two or three speakers in a conference then this is a good architecture. As diagram below represents a floor control scenario in RMTP type of architecture.

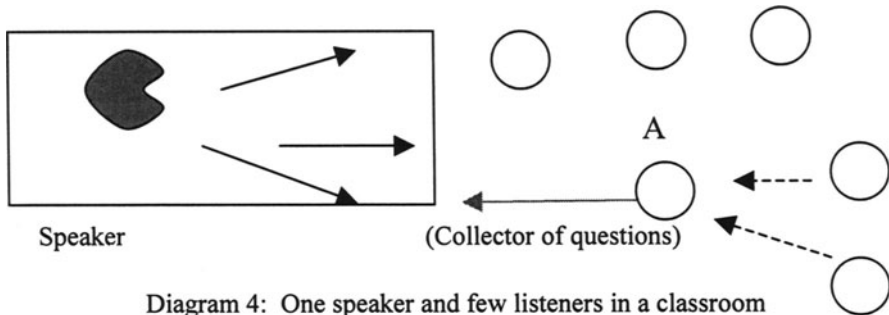


Diagram 4: One speaker and few listeners in a classroom

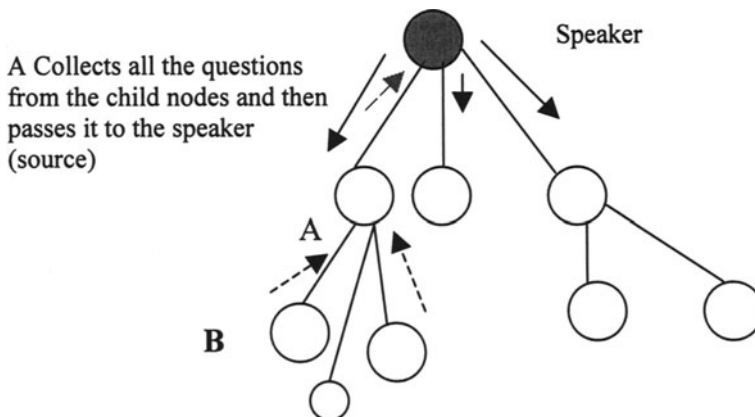


Diagram 5: Tree based architecture (e.g. RMTP) represents the classroom type of conferences

## 2.4 RLC and RMDP

In Reliable Multicast data Distribution protocol (RMDP), the problem of insuring reliable data delivery to large groups, and adaptability to heterogeneous clients is solved by Forward Error Correction (FEC) technique based on erasure codes[Vicisano98].

The basic principle behind the use of erasure codes is that the original source data, in the form of a sequence of  $k$  packets, along with additional  $n$  redundant packets, are transmitted by the sender, and the redundant data can be used to recover lost source data at the receivers. A receiver can reconstruct the original source data once it receives a sufficient number of ( $k$  out of  $n$ ) packets. The main benefit of this approach is that different receivers can recover from different lost packets using the same redundant data. In principle, this idea can greatly reduce the number of retransmissions, as a single retransmission of redundant data can potentially benefit many receivers simultaneously.

In order to deal with *congestion control*, the ultimate problem of one-to-many data transfer protocols on top of the IP multicast, RLC (receiver driven layered congestion control) is proposed by the same authors. This mechanism is designed for a transmitter sending data to many receivers on the Mbone[Levine98]. In Unicast communications, the sender takes part to congestion control by changing its sending rate according to the congestion signal that it receives. In multicast communications, this approach would be problematic, since different groups of receivers with different requirements exist, and adapting to the need of one set of receivers will be unfair to the rest. The effect of congestion control is decided by the receivers. It gives receivers the possibility to modulate the receive rate by joining/leaving layers.

Though the above mechanisms are very good solution for bulk data transfer, it does not really satisfy the needs for floor control. For example, in floor control mechanism the identity of the participants are quite crucial. Combination of RLC + RMDP is not really appropriate for floor control purposes.

## 2.5 PGM

Pretty good multicast (PGM) is a reliable transport protocol for applications that require ordered, duplicate free, multicast data delivery from multiple sources to multiple receivers[speakman98]. When a receiver detects a missing packet, it repeatedly unicasts a NAK to the last-hop PGM network element on the distribution tree from the source. A receiver repeats this NAK until it receives a NAK confirmation (NCF) multicast to the group from that PGM network element. The network element repeatedly forwards the NAK to the upstream PGM network element on the reverse of the distribution path from the source of the original data packet until it also receives an NCF from that network element. Finally, the source itself receives and confirms the NAK by multicasting an NCF to the group.

PGM is not intended for use with applications that depend either upon acknowledged delivery to a known group of recipients, or upon total ordering amongst multiple sources. For floor control, these two functionalities are quite crucial, therefore PGM is not the best suited protocol for floor control. PGM is

better suited for applications in which members may join and leave at any time, and that are either insensitive to unrecoverable data packet loss or are prepared to resort to application recovery in the event.

## 2.6 Functional Criteria

The table below is a comparison of several multicast transport protocols based on functions that are relevant for floor control.

<i>Protocol</i>	<i>Reliability Semantics</i>	<i>Con- gestion Control</i>	<i>Participant structure</i>	<i>Knowledge of participant</i>	<i>ACK/ NACKs/ Retrans- mission</i>	<i>Unit of delivery</i>
SRM	Reliable	No	Distributed	Via session messages	NAK, receiver reliable	1 ADU = app. Data unit
RMTP (BELL Labs)	Reliable	Yes	Hierarchy of regions, domain regions	Optional, may be known	Window of packets ACK/HACK	N = window size
RLC + RMDP	Reliable	Yes	No	No	No ACK/NAKs FEC for error recovery	K/N = depend on file size
PGM	Reliable	No	Local retransmitters	No	Bread crumb	1 packet
MTP/SO	Reliable, totally ordered, atomic delivery	Through different streams	Master Repeater Consumer	Known	NAK (?)	?
NTE	Reliable	No	Distributed	Via Session packets	Trig-gered NAKs with randomi-sation + FEC	1 ADU = 1 packet



### 3 FLOOR CONTROL AND ITS REQUIREMENTS

Floor control in CSCW is a metaphor for "assigning the floor to a speaker", which is not only applicable to voice channels, but more generally to any kind of sharable resource within conferencing and collaboration environments[Dommel95]. A floor is an individual temporary access or manipulation permission for a specific shared resource, e.g., a telepointer or voice-channel, allowing for concurrent and conflict-free resource access by several conferees.

For example, a floor requester in a meeting room would be a person who raises his/her hand up to ask a question. It is up to the chair to grant the floor to the requester. The session parameter entails the number of collaborators, and their role (chair, listener, a floor holder), determining their capabilities. Also, the interconnectivity (1-1, 1-to many, many-to-many), sharing distribution range(local, wide area), and link types (bi or unidirectional) are important too.

There are several types of floor control policy available for use by collaborative environments like explicit release, free floor, round-robin scheme etc.[Greenburg 91]. Whatever the scheme is, for applications to scale beyond a few participants, all communication must be multicast. Some research has been carried out to support Interactive collaboration application like TMTP[Sudan95] for data , STORM[Xu97] for audio and video and SRM[Floyd95] for wb. However, the nature of floor control is somehow different to these interactive applications. For example, the volume of data i.e. floor control messages are lot less than audio or video or whiteboard associated data, the timing of requesting/granting floor control can be very specific (for example, when the chair/speaker addresses the audience and asks for questions, a lot of listeners are going to request the floor but before that traffic may be lot less), ordering of data is more crucial factor than audio/video(for fairness, or applications like when customers are bidding for share) etc.

Typically traffic control for floor requests would be done in low level per source. An example of sudden flood of traffic would be "Flash Call" problem in POTS. Flash call would occur when a televoting system is taking place, where the viewers call a telephone number provided by a particular program, to give their opinion. The first method to avoid this sort of problem is the undeterministic approach, where after certain calls being taken by the network, users would hear an equipment engaged tone. This would stop the network being flooded by too many calls. Other approach is the deterministic approach, where the telephone company would be warned in a day advance, by the program organisers. So the telephone company can provide enough resources for that sort of service, and the cost would be higher.

On a data network, similar situation can take place too. There are certain traffic problems which would only apply to floor control and conference control type of applications. A reliable IP multicast protocol has to include certain features which would account for:

*Congestion control* - The volume of traffic will increase at certain points of time. The reliable IP multicast has to cope with sudden burst of traffic. Many sessions have precise starting times, when most of the members of a conference joining the session, or multimedia tools such as vat and vic can be programmed to join a session at the instant of its inception. This will cause a flood of traffic.

*Ordering* - To be fair to all the floor requesters the IP multicast has to have a mechanism for strict ordering. Let us consider if a receiver A requested for a floor who is 120 ms from the server/chairman. Receiver B requests for a floor who happens to be 100 ms away from the receiver/chairman after 10ms. Therefore, B's request will get to the server/chair before A's request, which is unfair for A. In this particular case the timing difference is so small that it may not really matter, but the difference can be in seconds rather than milliseconds.

*Reliability* - To provide good services, reliability and the retransmission strategy is quite important. Assume the scenario, where a floor request is multicast by a receiver A, receiver B didn't receive the message after time  $t$ . Receiver B now bids for the floor, without knowing the floor requester is Receiver A. Imagine there is a policy in this conference that if someone has requested a floor, the next person is not allowed to bid for the floor within next  $t'$  seconds. Now somehow in this scenario, someone has to inform the receiver B that receiver A has asked for the floor, and he may not request/being granted the floor. In protocols like TMTP the domain manager retransmit the data, whereas in SRM the nearest receiver to B will transmit the data.

*Member Classes* - There can be different types of members in a conference. As discussed in section 2.2, the rate controlled transmission of user data is very useful for floor control. For limited bandwidth, this is a way to limit number of concurrent users on the network. For example, one type of member will be not just a member but also a potential co-ordinator and repeater. Another type of members will be just normal members, the last type of member will unreliable receiver who will not ask for retransmission. If the members are categorised like that then the job of the application programmer is made a lot easier. A model like MTP/SO proposes to meet this requirement.

#### 4 LIMITATIONS OF FLOOR CONTROL

A lot of the multicast transport protocols like SRM, RMTP, MTP/SO will meet some of the requirements for floor control. Certain protocols can be customised or adopted to meet some of the requirements. However, there are some limitations of a floor control mechanism itself because of the nature of its behaviour. The principle difficulty is in achieving scalability to large group sizes. In a conference, where all members have access to the ability to request (and grant) the floor, it is necessary for all participants to know who the other participants are. Otherwise, none can see a global reason for giving someone the floor.

If the access bandwidth is small compared to network backbone bandwidth, at time  $t$ , there may be 1000 receivers in the system, however using RTCP the report

of the participants may show only first 20 participants<sup>\*</sup>. To account for congestion control a solution has been suggested in timer reconsideration for enhanced RTP scalability [Rosenberg98]. In a multimedia session which is using RTP/RTCP for transporting audio and video where RTCP rate is 1 kb/s. If all RTCP packets are 1 kb, packets should be sent at a total rate of one per second. Under steady state conditions, if there are 100 group members, each member will send a packet once every 100 seconds. However, if 100 group members all join the session at about the same time, each thinks they are initially the only group member and sends a packet at a rate of 1 per second, causing a flood of 100 packets per second or 100 kb/s, into the group.

So the effect of timer reconsideration algorithm is to reduce the initial flood of packets, which occur when a number of users simultaneously join the group. A participant P who wants to join at time  $t$  will determine the group size and it will transmit at time  $t'$ , where  $t' > t$ . So if a session has to start at 10:00 am, packets will be sent at 10:01 am, 10:02 am and so on. Therefore, at time  $t$ , the report showing the number of participants at 10:00 am will not be correct.

So the underlying technology has to support users to join a session at  $t''$  where  $t'' < t$ . In other words, if the session is programmed to be broadcast at 10:00 am, users have to join the session from 9:55 am. That requires modification of connection charges to include the traffic flow pre session.

If each participant sends messages at the rate of  $K/N$  per second, where  $K$  is the fraction of total capacity allowed for the RTCP messages, the following can be derived:

For audio, we might choose to have 1 speaker and therefore  $K$  is the capacity of that 1 flow. Typically RTCP messages might be limited to 5% of the flow, so for 20 packets per second, we would be allowed 1 message per second. Over 5 minutes, this would allow  $N$  to reach 300.

For video, we may choose to allow either one video flow or several, to save bandwidth, we probably choose the current speaker's video channel, we might be sending 100 packets per second from each and every source, which allows for  $K=5$ , or  $N$  to reach 1500 participants after 5 minutes.

## 5 OBSERVATIONS

Many protocols are proposed and implemented:

Protocols differ widely in design

Logical structure of communication pathways (ring versus tree versus none)

---

\* If the reliable protocol is distributed (e.g. in SRM/NTE) i.e. the participants can only see the local information straight away and overall statistics is an option, then this problem can be eliminated to an extent.

Group membership mechanisms and assumptions  
 Receiver-reliable versus sender reliable  
 ACK/NAK and FEC

Based on floor control requirements from a reliable IP multicast (as discussed in section 3.0) SRM will be one of the most suitable transport protocols if all the participants are multicast capable. Because SRM represents a simple and robust approach for large-scale recovery based on persistent state, suppression of duplicate NACKs and repairs, and global retransmissions. The messages specify a time-stamp used by the receivers to estimate the delay from the source, which causes global ordering. Also the model of this algorithm is distributed so that the participants list will not take too long to update. However, if the number of participants is very large, the convergence time will grow exponentially and SRM will not be the best suited algorithm.

If some of the participants in a video conference is Unicast only a tree based structure for IP multicast like RMTP or MTP/SO will be quite suitable too. In the hierarchical system, one parent node can have several Unicast only child nodes underneath it and it can Unicast the data to these child nodes. In this model the participants list can be viewed by the parent node as shown in diagram 5.

## 6 LIMITATIONS OF SRM AND MTP/SO

SRM is very efficient for retransmitting the lost packet whereas MTP is customised to take care of different classes of members in a conference. None of these protocols cater for congestion or flood of packets which will be caused by a session starting or question time for a conference for example. This sort of problem is solved RMDP or the approach taken by RTP timer reconsideration.

## 7 IDEAL PROTOCOL FOR CONFERENCE CONTROL

After discussing the pros and cons of the different protocols it seems that a reliable multicast protocol has to be able to provide:

Congestion control: Cope with sudden burst of traffic. If number of receivers are small (for example, if it is up to 100 receivers) a buffer can be provided to store the requests. Otherwise, a mechanism has to be provided where pre session traffic flow is allowed. RTP timer reconsideration is an example to deal with congestion control. Also if a user who just got the floor waits a certain amount of time before asking for the floor again will help the implosion as well.

Ordering: The point about floor control is that requestors should get a fair chance at getting the floor. The problem with the reliable multicast transport protocols is that to scale, they use techniques like SRM (random timer). What is required is a deterministic (round robin) timers for people requesting the floor at the same time. None of the transport protocols include this feature. So if a participant asked for the floor or got the floor last time, then they have to go after everyone else - i.e. that user/participant has to wait before asking for the floor again.

Reliability : Fastest way to retransmit lost/damaged packets. Not just the source, any one holding the packet will transmit the packet to the receiver require that damaged packet. SRM's retransmission strategy provides that.

Distributed control: As discussed in section 4.0, because convergence time increases as the number of users increase, there is a limit on the size of conference of known participants. A hierarchical system with just the knowledge of certain group or certain local users will be a possible solution. RMTP or STORM can provide that sort of architecture.

Simple: Multicast the status of the floor holders, a request is multicast to the group too. Any IP multicast can provide this function.

Other: May be able to cope with Unicast only receivers too. Security is provided for alternative approaches.

## 8 CONCLUSION

There are protocols like RMTP/STORM, NTE and SRM which are designed for specific applications. SRM is a robust protocol which meets a lot of the requirements for conference control. MTP and RMTP meet certain criteria too. However, these protocols need a level of customisation or a level of adaptation to be ideal protocol for conference control. This paper also looks at the limitation of these protocols and the limitation of floor control to achieve scalability. Therefore, if a reliable multicast has to be designed to meet the requirements of floor control it can be quite complicated to cater for ordering, congestion control, pre traffic flow etc. In order to keep it simple, we need a mechanism where the status of the floor holders is multicast in every few seconds to the group. If a user wish to bid for the floor, the request is multicast too. The stabilising time/converging time grows as the number of participants grow normally, so a hierarchical system will be a better solution. It is also required to provide a distributed model for retransmission and keep the status of receivers up to date.

## 6 REFERENCES

- Dommel P., Aceves JJ (1995) Floor Control for Activity coordination in Networked Multimedia Application - Proc. 2nd Asian-Pacific Conference on Communications (APCC)'95, Osaka, Japan, June 12-16, 1995.
- "FAQ - on the Mbone", <http://www.mediadesign.co.at/newmedia/more/mbone-faq.html>
- Floyd S., V. Jacobson, S. McCanne, C. G. Liu, and L. Zhang (1995) A Reliable Framework for Light-Weight Sessions and Application Level Framing - ACM SIGCOMM '95. Boston. August 30-September 1, 1995.
- Floyd S., Varadhan K., Estrin D (1998) Impact of Network dynamics on End-to-End protocols: Case studies in TCP and Reliable Multicast.- research draft "<http://www.isi.edu/~kawnan/VINT/ic98.ps>"
- ITU H.323 recommendation (1997)
- ITU draft recommendation T.120 (1997) - Data protocols for multimedia conferencing

- Levine B., Aceves-JJ (1998) A comparison of Reliable Multicast Protocols , "<http://www.ucsc.edu/b.levine>", Multimedia Systems (ACM/Springer), Vol. 6, No.5, August 1998.
- Lin, John C. and Paul, Sanjoy (1996) RMTP: A Reliable Multicast Transport Protocol, IEEE INFOCOM '96, March 1996, pp. 1414-1424.
- Ott J., Bormann C (1997) MTP/SO - Self organising Multicast Internet draft 1997, draft-bormann-mtp-so-01.txt
- Rosenberg J., Schulzrinne H.(1998)Timer Reconsideration for Enhanced RTP scalability - Internet draft -draft-ietf-avt-reconsider-00.ps
- Sudan M., R. Yavatkar, J. Griffefon (1993) A reliable Dissemination Protocol for Interactive Collaborative Applications, ACM multimedia
- Speakman T., Farincci D., Lin S.(1998) PGM specification - Internet draft draft-speakman-pgm-spec-00.txt
- Vicisano L., Rizzo L(1997) A Reliable Multicast Data distribution Protocol based on Software FEC techniques -Proceedings of the 4<sup>th</sup> IEEE workshop on the architecture and Implementation of High Performance Communication systems (HPCS 97)
- Vicisano L., Crowcroft J., Rizzo L.(1998) TCP Like congestion control for layered multicast data transfer - Proceeding of INFOCOM 1998
- X.Rex Xu, Zhang H, Yavatkar R (1997)Resilient Multicast Support for Continuous media applications "<http://research.ivv.nasa.gov/RMP/links.html>", in Proc. International Workshop on Network and Operating System Support for Digital Audio and Video(NOSSDAV), St. Louis, May 1997
- Zheng, W, Crowcroft J, Diot C. and Ghosh A (1997) Framework For Reliable Multicast Application Design, HIPPARCH 97.

## AUTOBIOGRAPHY

Nadia Kausar is a PhD student in Computer Science Department in University College London. Nadia is looking at various multiuser, multimedia system architectures over various networks and IP telephony for last 2 years. The work has initially focused on a comparison of the ITU's conferencing designed to run on circuit-switched networks and the IETF's MMUSIC conference control architectures with a view to extracting a generic set of modules to run over the Internet. Nadia graduated with a first class degree in Computer Systems in 1994 from University of Westminster. Nadia's main supervisor is Prof. Jon Crowcroft. This work is also supervised by Ian Marshall in BT Labs, Ipswich.

Jon Crowcroft is a professor of networked systems in the Department of Computer Science, University College London, where he is responsible for a number of European and US funded research projects in Multi-media Communications. He has been working in these areas for over 18 years. He graduated in Physics from Trinity College, Cambridge University in 1979, and gained his MSc in Computing in 1981, and PhD in 1993. He is a member of the ACM, the British Computer Society and a Fellow of the IEE and a senior member of the IEEE. He is a member of the IAB and general chair for the ACM SIGCOMM. He is also on the editorial team for the ACM/IEEE Transactions on Networks. With Mark Handley, is the co-author of WWW:Beneath the Surf (UCL Press); he also authored Open Distributed Systems (UCL Press/Artech House).