Internet Engineering Task Force
INTERNET-DRAFT

M. Handley/J. Crowcroft/C. Bormann
UCL/UCL/Bremen
22 Feb 1996

The Internet Multimedia Conferencing Architecture

Abstract

This document provides an overview of multimedia conferencing on the Internet. The protocols mentioned are all specified elsewhere as internet-drafts or RFCs. Each RFC gives details of the protocol itself, how it works and what it does. This document attempts to provide the reader with an overview of how the components fit together and of some of the assumptions made, as well as some statement of direction for those components still in a nascent stage.

This document is a product of the Multiparty Multimedia Session Control (MMUSIC) working group of the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at confctrl@isi.edu and/or the authors.

Status of this Memo

1 Introduction

In conjunction with computers, the term “conferencing” is often used in two different ways: firstly, to refer to bulletin boards and mail list style *asynchronous* exchanges of messages between multiple users; secondly, to refer to *synchronous* or so-called “real-time” conferencing, including audio, video, shared whiteboards and other applications. This document is about the architecture for this latter application, in the Internet. There are other infrastructures for conferencing in the world: POTS (Plain Old Telephone System) networks often provide voice conferencing and phone-bridges, while the ISDN provides H.320[2] for small, strictly organised video-telephony conferencing.

The architecture that has evolved in the Internet is far more general as well as being scalable to very large groups, and permits the open introduction of new media and new applications as they are devised.

The determining factors of a conferencing architecture are communication in (possibly large) groups of humans and real-time delivery of information. In the Internet, this is supported at a number of levels. The remainder of this section provides an overview of this support, and the rest of the document describes each aspect in more detail.

In a conference, information must be distributed to all the conference participants. Early conferencing systems used a fan-out of data streams, e.g., one connection between each pair of participants, which means that the same information must cross some networks more than once. The Internet architecture uses the more efficient approach of *multicasting* the information to all participants (section 2).

Multimedia conferences require real-time delivery of at least the audio and video information streams used in the conference. In an ISDN context, fixed rate circuits are allocated for this purpose — whether their bandwidth is required at any particular instance or not. On the other hand, the traditional Internet service model (“best effort”) cannot make the necessary quality of service available in congested networks. New service models are being defined in the Internet together with protocols to *reserve* capacity in a more flexible way than that available with circuit switching (section 3).

In a datagram network, multimedia information must be transmitted in packets, some of which may be delayed more than others. In order that audio and video streams be played out at the recipient in the correct timing, information must be transmitted that allows the recipient to reconstitute the timing. A *transport protocol* with the specific functions needed for this has been defined (section 4).

The humans participating in a conference generally need to have a specific idea of the context in which the conference is happening, which can be formalized as a conference *policy*. Some conferences are essentially crowds gathered around an attraction, while others have very formal guidelines on who may take part (listen in) and who may speak at which point. In any case, initially the participants must find each other, i.e. establish communication relationships (conference *setup*, section 5). During the conference, some conference *control* information is exchanged to implement a conference policy or at least to inform the crowd of who is present (section 6).

In addition, *security* measures may be required to actually enforce the conference policy, e.g. to control who is listening and to authenticate contributions as actually originating from a specific person. In the Internet, there is little tendency to rely on the traditional "security" of distribution offered e.g.& by the phone system. Instead, cryptographic methods are used for encryption and authentication, which need to be supported by additional conference setup and control mechanisms (section 7).

The present version of this document does not yet describe the architectural considerations underlying the conferencing applications other than audio and video that have evolved in the Internet, e.g. Imm, Wb[1], Nt.

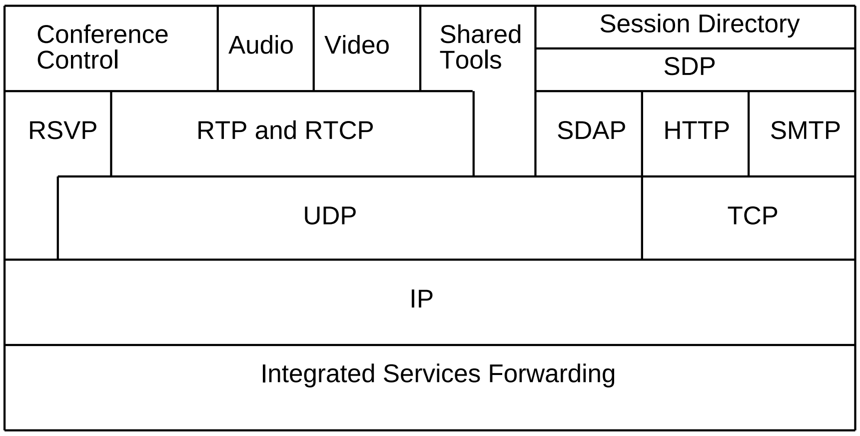

Figure 1: Internet multimedia conferencing protocol stacks

The protocol stacks for internet multimedia conferencing are shown in figure 1. Most of the protocols are not deeply layered unlike many protocol stacks, but rather are used alongside each other to produce a complete conference.

2 Multicast Traffic Distribution

IP multicast enables efficient many-to-many datagram distribution. It is one of the basic building blocks of the internet multimedia conferencing architecture. For most conferencing purposes, unicast is viewed as being a special case of multicast routing.

2.1 Multicast Service Model

The IP multicast service model is as follows:

- Senders send datagrams to the address of a *multicast group*.
- Receivers express an interest in (join) certain multicast groups.
- Multicast routers conspire to deliver multicast group addressed datagrams from the senders to the receivers.

The important factor here is that senders do not have to know who the receivers are in order to be able to send to them. In fact, in most situations, no single point in the network needs to know who all the receivers are, and it is this that makes IP multicast scalable to very large groups. In addition, receivers do not need to know who the senders are in order to be able to receive traffic from them, and this solves many conference setup and resource location problems without needing explicit machinery.

There are many multicast routing protocols [3],[4],[5],[6] but all of them satisfy the above service model. They differ in their mechanisms and in how they scale with the number of senders and groups.

Within a single LAN, group membership is expressed by IGMP[7]. IGMP version 3 allows receivers to express an interest in only receiving some of the senders to a particular multicast group. Earlier versions of IGMP only allow a receiver to request to receive all the sources sending to a multicast group.

2.2 Address Allocation

How does an application choose a multicast address to use?

In the absence of any other information, we can bootstrap a multicast application by using *well-known* multicast addresses. Routing (unicast and multicast) and group membership protocols[7] can do just that. However, this is not the best way of managing applications of which there is more than one instance at any one time.

For these, we need a mechanism for allocating group addresses dynamically, and a directory service which can hold these allocations together with some key (session information for example — see later), so that users can look up the address associated with the application. The address allocation and directory functions should be distributed to scale well.

Address allocation schemes should avoid clashes, hence some kind of hash function suggests itself for forming initial "random" values for the address. Furthermore, both the address allocation system and the directory service can take advantage of the baseline multicast mechanism by advertising conferences through multicast messages on a *well-known* address, and using this to inform other directory servers to remove clashes and inform applications of the allocation.

3 Internet Service Models

Traditionally the internet has provided best-effort delivery of datagram traffic from senders to receivers. No guarantees are made regarding when or if a datagram will be delivered to a receiver, however datagrams are normally only dropped when a router exceeds a queue size limit due to congestion. The best-effort internet service model does not assume FIFO queuing, although many routers have implemented this.

With best-effort service, if a link is not congested, queues will not build at routers, datagrams will not be discarded in routers, and delays will consist of serialisation delays at each hop plus propagation delays. With sufficiently fast link speeds, serialisation delays are insignificant compared to propagation delays.

If a link is congested, with best-effort service queuing delays will start to influence end-to-end delays, and packets will start to be lost as queue size limits are exceeded.

3.1 Non-best effort service

Real-time internet traffic is defined as datagrams that are delay sensitive. It could be argued that all datagrams are delay sensitive to some extent, but for these purposes we refer only to datagrams where exceeding an end-to-end delay bound of a few hundred milliseconds renders the datagrams useless for the purpose they were intended. For the purposes of this definition, TCP traffic is normally not considered to be real-time traffic, although there may be exceptions to this rule.

On congested links, best-effort service queuing delays will adversely affect real-time traffic. This does not mean that best-effort service cannot support real-time traffic — merely that congested best-effort links seriously degrade the service provided. For such congested links, a better-that-best-effort service is desirable.

To achieve this, the service model of the routers can be modified. At a minimum, FIFO queuing can be replaced by packet forwarding strategies that discriminate different "flows" of traffic. The idea of a flow is very general. A flow might consist of "all marketing site web traffic", or "all fileserver traffic to and from teller machines" or "all traffic from the CEOs laptop wherever it is". On the other hand, a flow might consist of a particular sequence of packets from an application in a particular machine to a peer application in another particular machine between specific times of a specific day.

Flows are typically identifiable in the Internet by the tuple: {source machine, destination machine, source port, destination port, protocol} any of which could be "ANY" (wildcarded).

In the multicast case, the destination is the group, and can be used to provide efficient aggregation.

Flow identification is called classification and a class (which can contain one or more flows) has an associated service model applied. This can default to best effort.

Through network management, we can imagine establishing classes of long lived flows — enterprise networks ("Intranets") often enforce traffic policies that distinguish priorities which can be used to discriminate in favor of more important traffic in the event of overload (though in an underloaded network, the effect of such policies will be invisible, and may incur no load/work in routers).

The router service model to provide such classes with different treatment can be as simple as a priority queuing system, or it can be more elaborate.

Although best-effort services can support real-time traffic, classifying real-time traffic separately from non-real-time traffic and giving real-time traffic priority treatment ensures that real-time traffic sees minimum delays. Non-real-time TCP traffic tends to be elastic in its bandwidth requirements, and will then tend to fill any remaining bandwidth.

We could imagine a future Internet with sufficient capacity to carry all of the world's telephony traffic. Since this is a relatively modest capacity requirement, it might be simpler to establish "POTS" as a static class which is given some fraction of the capacity overall, and then within the backbone of the network no individual call need be given an allocation (i.e. we would no longer need the call setup/tear down that was needed in the legacy POTS which was only present due to under-provisioning of trunks, and to allow the trunk exchanges the option of call blocking). The vision is of a network that is engineered with capacity for all of the average load sources to send all the time.

3.2 Reservations

For flows that may take a significant fraction of the network (i.e. are "special" and can't just be lumped under a static class), we need a more dynamic way of establishing these classifications. In the short term, this applies to any multimedia calls since the Internet is largely under-provisioned at the time of writing.

RSVP is being standardised for just this purpose. It provides flow identification and classification. Hosts and applications are modified to speak RSVP client language, and routers speak RSVP.

Since most traffic requiring reservations is delivered to groups (e.g. TV), it is natural for the receiver to make the request for a reservation for a flow. This has the added advantage that different receivers can make heterogeneous requests for capacity from the same source. Thus RSVP can accommodate monochrome, color and HDTV receivers from a single source.

Again the routers conspire to deliver the right flows to the right locations.

RSVP accommodates the wildcarding noted above.

3.3 Admission Control

If a network is provisioned such that it has excess capacity for all the real-time flows using it, a simple priority classification ensures that real-time traffic is minimally delayed. However, if a network is insufficiently provisioned for the traffic in a real-time traffic class, then real-time traffic will be queued, and delays and packet loss will result. Thus in an under-provisioned network, either all real-time flows will suffer, or some of them must be given priority.

RSVP provides a mechanism by which an admission control request can be made, and if sufficient capacity remains in the requested traffic class, then a reservation for that capacity can be put in place.

If insufficient capacity remains, the admission request will be refused, but the traffic will still be forwarded with the default service for that traffic's traffic class. In many cases even an admission request that failed at one or more routers can still supply acceptable quality as it may have succeeded in installing a reservation in all the routers that were suffering congestion. This is because other reservations may not be fully utilising their reserved capacity in those routers where the reservation failed.

3.4 Billing

If a reservation involves setting aside resources for a flow, this will tie up resources so that other reservations may not succeed, and depending on whether the flow fills the reservation, other traffic is prevented from using the network. Clearly some negative feedback is required in order to prevent pointless reservations from denying service to other users. This feedback is typically in the form of billing. For real-time non-best effort multicast traffic that is not reserved, this negative feedback is provided in the form of loss due to congestion of a traffic class, and it is not clear that usage based billing is required.

Billing requires that the user making the reservation is properly authenticated so that the correct user can be charged. Billing for reservations introduces a level of complexity to the internet that has not typically been experienced with non-reserved traffic, and requires network providers to have reciprocal usage-based billing arrangements for traffic carried between them. It also requires mechanisms whereby some fraction of the bill for a link reservation can be charged to each of the downstream multicast receivers.

4 Transport Protocols

So-called real-time delivery of traffic requires little in the way of transport protocol. In particular, real-time traffic that is sent over more than trivial distances is not retransmittable.

4.1 Separate Flows for each Media Stream

With packet multimedia data there is no need for the different media comprising a conference to be carried in the same packets. In fact it simplifies receivers if different media streams are carried in separate flows (i.e., separate transport ports and/or separate multicast groups). This also allows the different media to be given different quality of service. For example, under congestion, a router might preferentially drop video packets over audio packets. In addition, some sites may not wish to receive all the media flows. For example, a site with a slow access link may be able to participate in a conference using only audio and a whiteboard whereas other sites in the same conference may also send and receive video.

4.2 Receiver Adaptation

Best-effort traffic is delayed by queues in routers between the sender and the receivers. Even reserved priority traffic may see small transient queues in routers, and so packets comprising a flow will be delayed for different times. Such delay variance is known as jitter.

Real-time applications such as audio and video need to be able to buffer real-time data at the receiver for sufficient time to remove the jitter added by the network and recover the original timing relationships between the media data. In order to know how long to buffer for, each packet must carry a timestamp which gives the time at the sender when the data was captured. Note that for audio and video data timing recovery, it is not necessary to know the absolute time that the data was captured at the sender, only the time relative to the other data packets.

4.3 Synchronisation

As audio and video flows will receive differing jitter and possibly differing quality of service, audio and video that were grabbed at the same time at the sender may not arrive at the receiver at the same time. At the receiver, each flow will need a playout buffer to remove network jitter. Inter-flow synchronisation can be performed by adapting these playout buffers so that samples/frames that originated at the same time are played out at the same time. This requires that the time base of different flows from the same sender can be related at the receivers, e.g. by making available the absolute times at which each of them was captured.

4.4 RTP

The transport protocol for real-time flows is RTP[8]. This provides a standard format packet header which gives media specific timestamp data, as well as payload format information and sequence numbering amongst other things. RTP is normally carried using UDP. It does not provide or require any connection setup, nor does it provide any enhanced reliability over UDP. For RTP to provide a useful media flow, there must be sufficient capacity in the relevant traffic class to accommodate the traffic. How this capacity is ensured is independent of RTP.

Every original RTP source is identified by a source identifier, and this source id is carried in every packet. RTP allows flows from several sources to be mixed in gateways to provide a single resulting flow. When this happens, each mixed packet contains the source ids of all the contributing sources.

RTP media timestamp units are flow specific — they are in units that are appropriate to the media flow. For example, 8kHz sampled PCM encoded audio has a timestamp clock rate of 8kHz. This means that inter-flow synchronisation is not possible from the RTP timestamps alone.

Each RTP flow is supplemented by Real-Time Control Protocol (RTCP) packets. There are a number of different RTCP packet types. RTCP packets provide the relationship between the realtime clock at a sender and the RTP media timestamps, and provide textual information to identify a sender in a conference from the source id.

4.5 Conference Membership and Reception Feedback

IP multicast allows sources to send to a multicast group without being a receiver of that group. However, for many conferencing purposes it is useful to know who is listening to the conference, and whether the media flows are reaching receivers properly. Accurately performing both these tasks restricts the scaling of the conference. IP multicast means that no-one knows the precise membership of a multicast group at a specific time, and this information cannot be discovered, as to try to do so would cause an implosion of messages, many of which would be lost[1]. Instead, RTCP provides approximate membership information through periodic multicast of session messages which, in addition to information about the recipient, also give information about the reception quality at that receiver. RTCP session messages are restricted in rate, so that as a conference grows, the rate of session messages remains constant, and each receiver reports less often. A member of the conference can never know exactly who is present at a particular time from RTCP reports, but does have a good approximation to the conference membership.

Reception quality information is primarily intended for debugging purposes, as debugging of IP multicast problems is a difficult task. However, it is possible to use reception quality information for rate adaptive senders, although it is not clear whether this information is sufficiently timely to be able to adapt fast enough to transient congestion. However, it is certainly sufficient for Van Jacobson congestion control[10] style adaption to a "share" of the current capacity.

5 Conference Control

Conferences come in many shapes and sizes, but there are only really two models for conference control: light-weight sessions and tightly coupled conferencing. For both models, rendezvous mechanisms are needed. Note that the conference control model is orthogonal to issues of quality of service and network resource reservation. Note also that the issue of conference control is orthogonal to the mechanism for discovering the conference.

5.1 Light-weight Sessions

Light-weight sessions are multicast based multimedia conferences that lack explicit session membership and explicit conference control mechanisms. Typically a lightweight session consists of a number of many-to-many media streams supported using RTP and RTCP using IP multicast[2]. The only conference control information available during the course of light-weight sessions is that distributed in the RTCP session information, i.e. an approximate membership list with some attributes per member.

5.2 Tightly coupled Conferences

Tightly coupled conferences may also be multicast based and use RTP and RTCP, but in addition they have an explicit conference membership mechanism and may have an explicit conference control mechanism that provides facilities such as floor control.

[1] Note that a conference policy that restricts conference membership can be implemented using encryption and restricted distribution of encryption keys, of which more later.

[2] There is some confusion on the term session, which is sometimes used for a conference and sometimes for a single media stream transported by RTP. In this document, we prefer to use the less ambiguous term conference except where existing protocols use the term session.

In the internet community, no standard mechanism for performing tightly coupled conference control currently exists. At the time of writing, it seems likely that a protocol based on the ITU's T.124[11] recommendation will be derived for internet usage.

6 Conference Discovery

There two basic forms of conference discovery mechanism. These are session advertisement and session invitation. Session advertisements are provided using a session directory, and inviting a user to join a session is provided using a session invitation protocol.

6.1 Session Directories

The rendezvous mechanism for light-weight sessions is a multicast based session directory. This distributes session descriptions[9] to all the potential session participants. These session descriptions provide an advertisement that the session will exist, and also provide sufficient information including multicast addresses, ports, media formats and session times so that a receiver of the session description can join the session. The protocol SDP (session description protocol) describes contents and format of the session descriptions.

As dynamic multicast address allocation can be optimised by knowing which addresses are in use at which times, the session directory is an appropriate agent to perform multicast address allocation. SDAP (session directory announcement protocol) is the protocol used by the session directory agents.

This mechanism can also be applied to advertised tightly coupled sessions, and only requires that additional information about the mechanism to use to join the session is given.

6.2 Session Invitation

Not all sessions are advertised, and even those that are advertised may require a mechanism to explicitly invite a user to join a session. Such a mechanism is required regardless of whether the session is a lightweight session or a more tightly coupled session, although the invitation system must specify the mechanism to be used to join the session.

As users are mobile, it is important that such a mechanism is capable of locating and inviting a user in a location independent manner. This requires that an extra level of indirection (addressing) is required from that provided by MMCC [13]. The invitation mechanism should also provide for alternative responses, such as leaving a message or being referred to another user, should the invited user be unavailable.

Based on a protocol with many of the properties required[12], a session invitation protocol (SIP) is being developed.

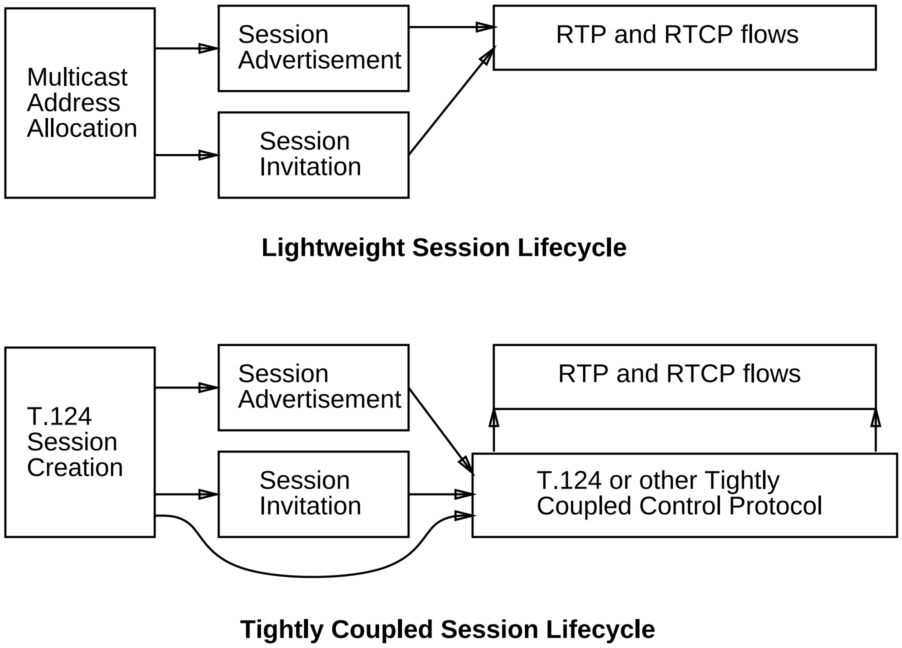

Figure 2: Internet multimedia conferencing lifecycles

7 Security

There is a temptation to believe that multicast is inherently less private than unicast communication since the traffic visits so many more places in the network. In fact, this is not the case except with broadcast and prune type multicast routing protocols[4]. However, IP multicast does make it simple for a host to anonymously join a multicast group and receive traffic destined to that group without the other senders' and receivers' knowledge. If the application requirement (conference policy) is to communicate between some defined set of users, then strict privacy can only be enforced in any case through adequate end-to-end encryption.

RTP specifies a standard way to encrypt RTP and RTCP packets using private key encryption schemes such as DES[14]. It also specifies a standard mechanism to manipulate plain text keys using MD5[15] so that the resulting bit string can be used as a DES key. This allows simple out-of-band mechanisms such as privacy-enhanced mail to be used for encryption key exchange.

7.1 Authentication and Key Distribution

Key distribution is closely tied to authentication. Conference or session directory keys can be securely distributed using public-key cryptography on a one-to-one basis (by email, a directory service, or by an explicit conference setup mechanism), but this is only as good as the certification mechanism used to certify that a key given by a user is the correct public key for that user. Such certification mechanisms[16] are not specific to conferencing, and no standard mechanisms are currently in use for conferencing purposes other than

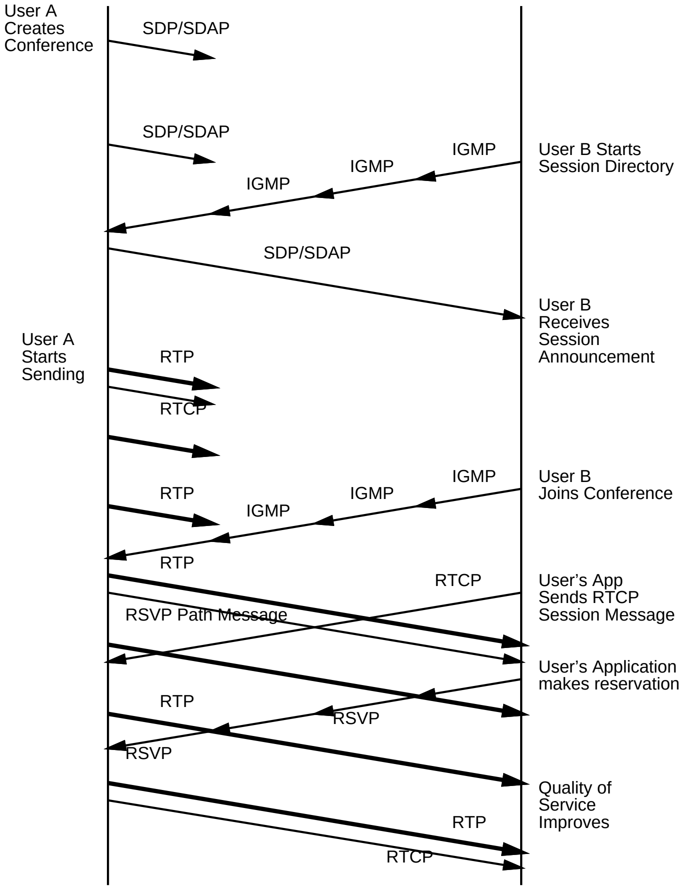

Figure 3: Joining a light-weight multimedia session

PEM[17].

At the time of writing, no standard mechanisms for key distribution are defined for the conference setup and control protocols in use.

Even without privacy requirements in the conference policy, strong authentication of a user is required if making a network reservation results in usage based billing.

7.2 Encrypted Session Announcements

Session Directories can make encrypted session announcements using private key encryption, and carry the encryption keys to be used for each of the conference media streams in the session. Whilst this does not solve the key distribution problem, it does allow a single conference to be announced more than once to more than one key-group, where each group holds a different session directory key, so that the two groups can be brought together into a single conference without having to know each other's keys.

8 Summary

This document is an attempt to gather together in one place the set of assumptions behind the design of the Internet Multimedia conferencing architecture, and the services that are provided to support it.

Figure 3 shows the time sequence involved in setting up a light-weight session between two sites. In this case, site A creates a session advertisement, and some time later starts sending a media stream even though there may be no receiver at that time. Some time later, site B joins the session, and starts to receive the traffic. At the earliest opportunity site B also makes an RSVP reservation to ensure the flow quality is satisfactory.

9 Acknowledgments and Authors Address

Acknowledgments are due to the End-to-End Research Group, the Int-serv, RSVP, MMUSIC and AVT working groups of the IETF, and discussion with colleagues at UCL. The earliest clear exposition of the ideas here can be found at http://www-mice.cs.ucl.ac.uk/mice-old/van/ and was presented at ACM SIGCOMM 1994 in London by Van Jacobson.

Mark Handley, Jon Crowcroft,
Department of Computer Science
University College London
Gower Street,
London WC1E 6BT
UK
fax +44 171 387 1397
Email: m.handley@cs.ucl.ac.uk, j.crowcroft@cs.ucl.ac.uk
Web: http://www.cs.ucl.ac.uk/index.html

Carsten Bormann
Universitaet Bremen
Postfach 330440
D-28334 Bremen GERMANY
fax +49 421 2038097
Email: cabo@informatik.uni-bremen.de
Web: http://www.informatik.uni-bremen.de/cabo

References

[1] "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing" S. Floyd, V. Jacobson, S. McCanne, C-G. Liu, L. Zhang ACM SIGCOMM 1995, pp 342-356

[2] ITU Recommendation H.320

[3] "An Architecture for Wide Area Multicast Routing" S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C-G. Liu, L. Wei ACM SIGCOMM 1994, London October 1994, ACM CCR Vol 24, No. 4, 126-135

[4] RFC 1075 S. Deering, C. Partridge, D. Waitzman, "Distance Vector Multicast Routing Protocol", 11/01/1988.

[5] "An Architecture for Scalable Inter-Domain Multicast Routing", A. Ballardie, P. Francis, J. Crowcroft ACM SIGCOMM 1993, pp 85-95

[6] RFC 1584 J. Moy, "Multicast Extensions to OSPF", 03/24/1994.

[7] Steve Deering "Multicast Routing in Internetworks and Extended LANs", ACM SIGCOMM 88, August 1988, pp 55-64 and Host Extensions for IP Multicasting, RFC 1112

[8] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson "RTP: A Transport Protocol for Real-Time Applications" Internet Draft draft-ietf-avt-rtp-08.txt, Work In Progress, Late 1995.

[9] M. Handley, V. Jacobson "SDP: Session Description Protocol" Internet Draft draft-ietf-mmusic-sdp-01.txt, Work in Progress, Nov 1995.

[10] V. Jacobson "Congestion Avoidance and Control", ACM SIGCOMM 88, August 1988

[11] ITU Recommendation T.124 — Generic Conference Control

[12] Schulzrinne, H., "Personal Mobility for Multimedia Services in the Internet" IMDS'96, March 4-6 1996. ftp://ftp.fokus.gmd.de/pub/step/papers/Schu9603:Personal.ps.gz

[13] Schooler, E., A Distributed Architecture for Multimedia Conference Control, ISI Research Report ISI/RR-91-289, November 1991. ftp://ftp.isi.edu/pub/hpcc-papers/mmc/mmcc.ps

[14] National Institute of Standards and Technology (NIST), "FIPS Publication 46-1: Data Encryption Standard", January 22, 1988

[15] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, MIT Laboratory for Computer Science and RSA Data Security, Inc., April 1992

[16] CCITT (Consultative Committee on International Telegraphy and Telephony). "Recommendation X.509: The Directory — Authentication Framework." 1988.

[17] RFC 1421 J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", Feb 1993