Personal Mobility for Multimedia Services in the Internet*

Henning Schulzrinne — GMD Fokus, Berlin — schulzrinne@fokus.gmd.de

Frank Oertel — TU Berlin

Christian Zahl — TU Berlin — cz@fokus.gmd.de

October 9, 1995

Abstract

Personal mobility is one of the goals of Universal Personal Telecommunications (UPT) being specified for future deployment. Most current efforts focus on telephony, with SS7 signaling. However, many of the same goals can be accomplished for multimedia services, by using existing Internet protocols. We describe a multimedia call/conference setup protocol that provides personal videophone addresses, independent of the workstation a called party might be using at the time. The system is set up to use the existing Internet email address as a videophone address. Location and call handling information is kept at the subscriber's home site for improved access and privacy.

1 Introduction

The use of multimedia has progressed from stand-alone applications, to point-to-point, mostly local applications such as video-on-demand or LAN-based video conferencing. Across the wide-area, low-bandwidth circuit-switched teleconferencing (such as those using the H.320 standard) are spreading, as well as simple multimedia file delivery in the World-Wide Web context. H.320 conferences use the call control facilities available for either ISDN or plain old telephony (POTS). H.320 conferences use multipoint control units (MCUs) for multiparticipant conferences and tend to scale badly with increasing numbers of conference participants.

IP multicast has been used experimentally in the Internet for both interactive conferencing and audio/video distribution [1, 2] for groups up to several hundred participants. For the MBONE, conferences and seminars are typically announced worldwide or regionally through a multicast session directory. While IP multicast

*This work was supported in part by the ACTS project Multicube (AC 422).

scales nicely to large groups, a multicast session directory is not suitable for thousands of concurrent phone calls or small-group conferences.

Recently, a number of companies have introduced applications and speech compression algorithms that allow personal computers to conduct voice conversations across the Internet even from modem-connected personal computers. These calls are point-to-point, audio only and assume that the called party resides at a known IP address. Generally, all Internet-based conferencing tools lack an easy-to-use mechanism to call up other users or invite them to a conference. That functionality is the subject of this paper.

We also intend to show that much of the control functionality envisioned for "intelligent networks" can be supplied by simple extensions of existing Internet services, running on workstations and personal computers. We describe a multimedia call control agent that offers flexible support for mobility and call processing. We use this to argue how computer-oriented intelligent networks should be structured for maximum flexibility and competitive service provision.

2 Multimedia Signaling and Mobility

Two kinds of mobility can be distinguished: *terminal mobility* and *personal mobility*. Terminal mobility allows to move a terminal[1] (telephone, workstation, laptop, PDA, etc.) from one location to another, while maintaining communication[2]. While terminal mobility is typically associated with wireless access, wired mobility, i.e., the ability to plug in a terminal at different locations, is also of interest.

Mobile IP efforts [3] provide the ability to move during a call without losing packet-level connectivity. Also, IP multicast [4] can be used to ensure continuous packet-level connectivity in a multimedia conference as long as the end system can listen and transmit in two cells simultaneously, so that it can join the multicast group in the new cell while still receiving data packets from the old cell.

"Personal mobility is the ability of end users to originate and receive calls and access subscribed telecommunication services on any terminal in any location, and the ability of the network to identify end users as they move. Personal mobility is based on the use of a unique personal identity (i.e., 'personal number')." [5, p. 44]. The issue of naming will be discussed in detail in Section 2.1.

Terminal and personal mobility are two aspects of the "intelligent network" envisioned by telecommunication network operators (PNOs). Some current services such as 800-numbers (free phone) and call forwarding are viewed as first-generation intelligent network services. Recently, the moniker Universal Personal

[1] or end system in Internet parlance

[2] Further distinctions can be made as to whether, say, on-going TCP connections or phone calls are kept up across moves, with or without data movement.

Telecommunication (UPT) has been used to describe the ability to enable communication with a person at any time, at any place, and in any form [5].

The work described here provides a form of personal mobility. The call mechanism assigns each subscriber a permanent address naming a "home base" that manages call handling and forwarding to the terminal the callee is currently using. That terminal may be wired or wireless. While the "home base" should be continuously reachable, the terminal may not be.

Currently, most residential PCs acessing the Internet via modem are only connected to the Internet a few minutes or hours a day, making it difficult to use the PC as a communication terminal for private use. In the longer term, continuously connected systems are desirable, with low-power standby PCs and continuous network connectivity. Continuous connectivity is particularly easy if access is provided by a shared medium like CATV, as lower-layer call setup can be avoided. For IP-over-ISDN, the router at the Internet service provider will automatically set up an ISDN call when the first signaling packet arrives.

2.1 Naming for Personal Mobility

The most visible manifestation of personal mobility the change from a telephone number as an identifier of a jack in the wall and a terminal[3] to a personal number, identifying a person. Thus, a person would maintain the same number even when moving or when switching from a wireline telephone to a mobile. 700 and 800 numbers in the US already have that property, although they are typically linked to a single wireline terminal for extended periods of time. However, it seems likely that numbers will remain national, so that a move across borders will require adopting a new identifier. While 800 numbers have recently been made portable across network providers, 700 numbers currently remain provider-specific.

Even if freed from its role as a terminal and location identifier, a telephone number remains a number, with all the disadvantages for human interaction:

- hard to memorize, even more so if any "predictable" components like area codes are removed;
- not guessable from a person's name, residence or professional affiliation[4];
- lack of redundancy leads to misdialed calls;
- area codes and other parts of the phone number are subject to wholesale changes;

[3]Even telephone numbers can be reassigned from one physical line to another, but usually only within a single exchange, unless expensive special arrangements are made.

[4]This "feature" obviously currently provides the only protection against nuisance calls - security through obscurity.

- different communication modes for the same person have different numbers (fax and phone, say);
- assignment in an environment with competitive local service is difficult;
- the number to be dialed depends on the call origin, e.g., the area code or country code must not be dialed for numbers within that area code or country.

On the other hand, the Internet has had a fairly usable naming mechanism for many years, in the form of the domain name system [6, p. 650ff]. Here, individual Internet hosts are designated by a hierarchical domain name, e.g., `ursa.fokus.gmd.de`. For electronic mail, RFC 821 specifies the familiar @ notation, with the form "name@host" or "name@domain". For private users without their own domain name, the email address currently contains the name of the service provider, e.g., `smith@aol.com`. Some professional organizations offer mail forwarders, so that one can maintain a single address, say, `j.doe@ieee.org`, despite changing employers or residence. Currently little used, there is also the possibility of geographical names, e.g., `j.smith@provider.amherst.ma.us`.

Initially, email addresses designated the Internet-attached host responsible for electronic mail and the user name on that host, leading to such email addresses as `bub9193@vax135.ho.att.com`. Many domains offer more friendly naming. First, various combinations of first and last name can be used, which are then mapped locally to a user account (login) name. Secondly, only the domain needs to be named, rather than the actual host responsible for receiving mail. The translation of domain names to the host name handling mail for a particular domain is carried out by the domain name system (DNS). So-called MX records map a domain name to a preference-ordered list of mail exchanger hosts. For example, the domain name `sun.com` maps to the hosts `mercury.sun.com` as the preferred mail host and `venus.sun.com` as a secondary one, should the first one be unreachable or busy. Equal-weight hosts can be specified for load sharing purposes. Compared to telephone numbers, "modern" email addresses can be easily remembered or even guessed. For example, knowing that somebody works at AT&T, it is sufficient to simple send e-mail to `somebody@att.com`. A list of candidate names and email addresses is returned should the name be ambiguous. Also, email addresses allow functional rather than personal designations, such as `info@fokus.gmd.de`. Even in computer-supported communication, the ability to memorize and pronounce addresses has proven to be important, as the virtual demise of the structured, but unwieldy X.400 addresses has shown. Unlike telephone numbers, the same email address can be used from anywhere in the Internet and it contains sufficient redundancy to likely fail outright rather than reach the wrong destination.

Email-style addresses have some drawbacks, in particular their larger size (compared to telephone numbers) and their limitation in character sets to letters, digits and hyphens. The use of Unicode may make it possible for those with accented names to be represented appropriately. However, this would further exacerbate the input problem, in that a much larger keyboard is already required than for numbers. This is a particular problem for mobile devices, but for messaging and other uses, some form of alphabetic input will have to be provided in any event.[5] For private use, it remains to be seen whether there is demand for provider-independent, "neutral", but by necessity geographically-based mail exchangers. It would also have to deal with the issue of name duplication within larger geographic locales. More realistically, some set of directory services might be used to map from common names and other attributes to a provider-specific name. Unfortunately, deployment and maintenance of Internet white pages (X.500 and otherwise) has not been a great success. It appears unlikely that a single directory hierarchy can be created, so that competing address services need to be searched.

2.2 Locating the Called Party

Implementing personal mobility requires locating the called party, more precisely, the workstation or other terminal that the callee currently has access to. Full personal mobility also requires the ability to locate the appropriate network or type of communication device. In the system described here, scripts set up by the user make that choice.

The method of locating a called party is independent of the call setup protocol and can be designed according to local performance and privacy requirements. Typically, the called party can only answer a multimedia call when logged into the console of a workstation, limiting the amount of information that needs to be tracked.

The interaction of the server handling incoming calls and the location server is a local matter, and can range from integrating the location server with the call server to a full CORBA-based location daemon operating in the local area network. Location servers can be either centralized, with the attendant increased failure probability and load concentration problems, or distributed. Location servers can use active badges [7], or more traditional Unix services such as rwho, rusers or finger[6].

User location can be *on-demand*, *polled* or *event-driven* plus a combination of these. In on-demand user location, the location server only tries to determine the user's location when a request arrives. In a polled system, the location server maintains a user location database. Finally, in an event-driven system, users logging in and out send location updates to one or more location servers. On-demand user location is advantageous

[5] In the U.S., at least, there have been efforts from the beginning of dial telephony to use names instead of numbers. American telephones have letters printed next to each digit; corporations often advertise spelled-out names (1-800-CALL-ATT) rather than numbers. Indeed, early on, the numbers of exchanges used to be based on their names.

[6] Unfortunately, rusers and rwho use data link layer broadcasts and are thus usually disabled

if calls are less frequent than user login changes or moves. They are also necessary if a user is logged onto several different workstations at the same time, and the workstation where the user has been most recently is to be located.

An effective method of demand-driven user location multicasts a search request when a call arrives. Login names are hashed into one of a range of multicast addresses. Each host subscribes only to the multicast group to which the user logged in the console belongs, thus significantly reducing its processing load for location requests. This method requires a daemon which, on every host, interacts with the login process or periodically checks the `utmp` file and responds to multicast queries.

In many cases, the number of workstations a user is likely to frequent within her domain is rather small. Thus, without any additional support on workstations, login location information can be guessed at by having the finger protocol randomly probe hosts based on past experience, noting the last login-from terminal.

3 The MUCS Protocol

In this section, we describe the basic mechanism used for establishing and changing multimedia calls and changing their parameters. The MUCS (multimedia conferencing system) protocol allows great flexibility in the location of decision making. Because of the advantages spelled out in Section 2.1 and because it is already widely available from network directory services to business cards and advertisements, the protocol uses the standard electronic mail address for reaching potential multimedia call participants. However, it is possible to use a different address, using the same format. We will generically refer to the `name@host`-style address for multimedia conversations as the *videophone address*.

The protocol follows the standard Internet client-server model, with the caller acting as client. Like many other Internet protocols including SMTP [8] (electronic mail), NNTP [9] (network news), HTTP [10] (WWW), ftp and telnet, the MUCS protocol consists of requests and replies written as ASCII text lines. The requests begin with a one-line command, as described below. The replies start with the protocol identifier and version, followed by a numeric status code and an optional reason phrase. Status codes have three digits, with the first digit indicating an information response (1), success (2), redirection (3), a client error (4) or a server error (5). A client does not have to understand a particular code, but can simply use the first digit to detect success or failure. Request and reply may be followed by a sequence of type-value headers, terminated by a blank line. The request and reply formats, as well as other protocol details, are modeled on the Hypertext Transfer Protocol (HTTP) [10]. Currently, protocol requests are not pipelined [11], as each caller is likely to generate only one request.

Naturally, the basic protocol could be implemented in a binary encoding, such as ASN.1, but request rates are expected to be low enough that the advantages of easy generation and parsing by programs such as Tcl [12] and humans outweigh efficiency costs.

3.1 Call Setup

Two-party calls and invitations to multicast conferences proceed in the same manner. During a multicast conference, additional conferees can be invited by any participant with the same mechanism. The call proceeds in two stages. First, the caller's client locates a MUCS server for the called party, and then sends the call request. A flow chart for locating a MUCS server is shown in Fig. 1. The sequence of lookups is designed to work even when only a few systems implement the MUCS protocol and makes maximum use of the email infrastructure for obtaining current location information. DNS address (A) records map a host name to one or more Internet addresses, while mail-exchange (MX) records [13], as mentioned, map a domain name to the host name of its mail exchange host. Rather than using MX records and thus forcing MUCS servers to be co-located with mail exchangers, a new service (SRV) resource record [14] can be used, once it is implemented more widely. However, using the mail host as a MUCS server has practical advantages. Due to the importance of electronic mail, it is likely to be well cared for and reliable. It is also often the only host that resides outside a corporate firewall.

If no server can be found or the hosts do not accept MUCS connections, the client attempts to map the videophone address to a more current one with the verify (VRFY) SMTP command or to expand a group alias to a list of addresses with the EXPN command.

If no working MUCS server can be located, a MIME mail message of type `application/mucs` is sent and the call attempt ends. Since mail delivery can be slow, taking from minutes to hours[7], there is no success/failure indication to the caller in this case. If the call request contains a multicast address, the callee simply joins that group whenever she reads the mail message, assuming the life time of the conference (see below) has not expired. For unicast calls, the recipient of the mail message in turn initiates a MUCS call.

If a working MUCS server has been found, a simple protocol exchange

- offers hints on the subject and urgency of the call;
- determines the type of media to be used during the call;
- establishes common media encodings;

[7]Particularly if an X.400 system is involved.

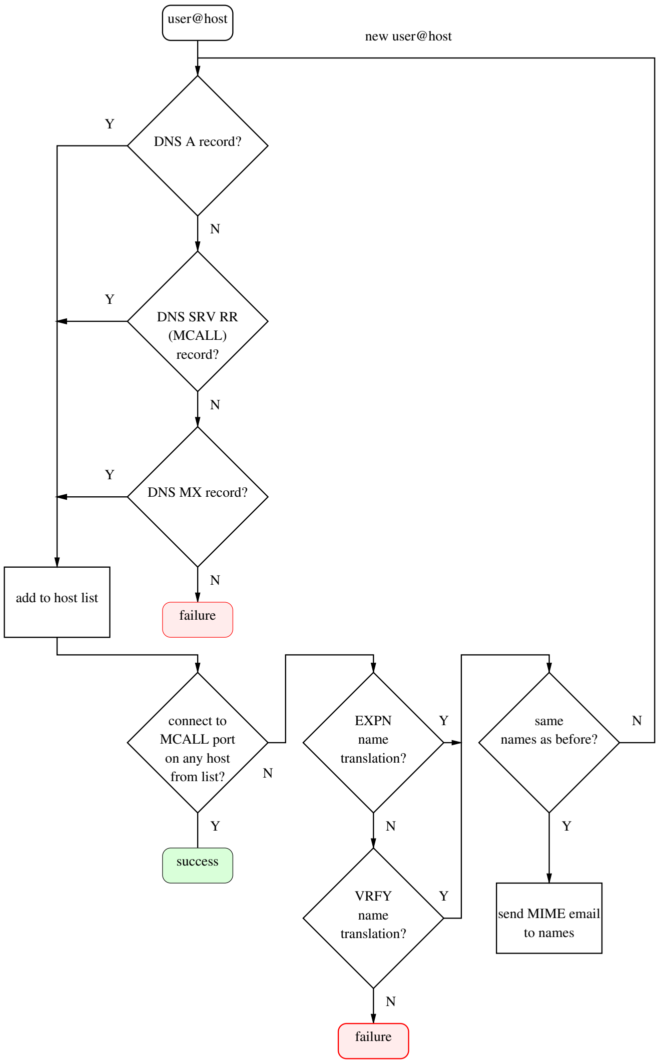

Figure 1: Mapping a videophone address to a MUCS server

- provides encryption keys for media.

The MUCS protocol currently uses TCP to a MUCS-specific well-known port to reliably exchange requests and replies. Handshake delays can be reduced by transactional TCP [15].

It could be argued that the call setup protocol should contact callees via IP multicast. However, this implies the need for a reliable multicast protocol, an effort probably not worthwhile given the typically low number of targets. It also has a bootstrapping problem, in that the called parties (and only those) have to subscribe to a common multicast group.

A sample call request in shown in Fig. 2. The first line indicates the desired action action, the name part of the videophone address, and the protocol version. This request line is followed by a number of header items indicating, for example, the caller name (From), user agent software (User-Agent) and call priority (Priority). The Call-Id, formatted like e-mail message identifiers, is used to identify a particular call for later changes (Section 3.2). The To field indicates that this particular call has been forwarded from an original group destination to several individual names. It has been referred to this host (in proxy mode, see below) by `ceres.fokus.gmd.de`.

Media are listed as Required or Accept headers. Each such header represents one media, with alternatives within each media listed in decreasing order of preference. Media descriptions uses the MIME grammar[8]. Each media entry lists the type (audio, video, application, …), the encoding and a list of parameter/value pairs, including the port number, the transport protocol, the RTP payload type [17], any floor control properties and the bit rate to be reserved. If the callee does not support at least one of the media listed in a Required header, it returns "None available" and the call fails. In the example, the caller is only interested in a call if the callee can communicate in one of the three audio formats listed, with PCMU (μ-law PCM), one channel, 16000 Hz sampling rate, preferred. The callee must also support either video at 128 or 250 kb/s, or, if all else fails, a shared whiteboard application. A shared editor is also a possibility, if the callee wishes to use it.

If the callee wishes to use a media, it returns Accept headers, in the order sent by the caller, and listing all acceptable encodings. (Listing all acceptable encodings rather than just the preferred one is necessary to allow the caller to contact called parties in turn and successively narrow the range of possible encodings.)

The protocol only implements a one-round negotiation for media types. If more complicated negotiations, including voting, with several participants are required, a more sophisticated protocol such as the agreement protocol [18] must be used. However, it seems likely that most applications will support a range of encodings, with large common subsets, so that elaborate negotiations may not be necessary. Applications are free to switch encodings during the call to any one of those accepted, for example to adapt to changing network congestion conditions or quality requirements [19]. We assume that applications can receive different media types concurrently and that the media type sent does not have to equal the media type received. This is generally true for today's MBONE applications[9], but may not be true for hardware devices.

If the callee has moved permanently, e.g., after leaving a company, the server returns status code 301, "Moved permanently", and provides a list of possible new locations, if known, as Location headers. The calling application will likely indicate a failure and possibly allow updating of a local phone directory. If the user has left without leaving a forwarding address, code 410 "Gone" can be returned. If the callee is at a different host, but only temporarily, status code 302, "Moved temporarily", is returned instead. This is one

[8]The SDP [16] format could have been used, but appears more cumbersome and breaks with Internet header conventions.
[9]With the exception of different audio sampling rates

of the mechanisms for providing personal mobility.

The callee (or her authorized software agent) can also indicate that she is temporarily unable or unwilling to take the call:

Request timeout: the called party could not be reached;

Busy: too many other calls or otherwise occupied;

Forbidden the caller is not allowed to reach the callee or the callee doesn't exist;

Identification required: A From header is required;

Authentication required: The request must be authenticated (see Section 3.5).

Optionally, the server can indicate a more opportune time for calling with the Retry-After header or can suggest a substitute with the Location header. For example, a response indicating two possible locations may look like this:

```
MUCS/1.0 302 Callee has moved temporarily
Location: jones@salt.lab3.company.com
Location: jones@pepper.lab3.company.com
```

At any time, either caller or callee can terminate the call attempt by closing the TCP connection. No explicit indication of ringing is provided. Note that the TCP connection is only open while ringing the callee, not during the actual call. A caller may also close the connection if it is trying to reach a user at several different locations simultaneously and gets the first positive answer.

A server can operate in two modes. In *proxy mode* (Fig. 3), the server tries to locate the called party locally, and only forwards the final result of its attempts to the client. For the local users, the first server acts like a caller and thus needs to implement both server and client operations, including loop detection. Proxy mode is appropriate for fire-walled companies, or when the location of users should not be revealed. In this mode, a tree of connections can be established, which must be torn down if any of the root links is severed. Proxy mode differs from the usual email message forwarding in that the original connection from caller to proxy remains open while the proxy contacts other hosts. This greatly simplifies error reporting and is rather more appropriate for an interactive medium. Proxy mode is similar to the resolutionn strategy employed by the X.500 or DNS directories.

In *redirect mode* (Fig. 4), each call server only provides further clues as to the location by issuing redirects. The client can then decide which, if any, of the locations to contact. The client is also responsible

```
CALL hgs@lupus.fokus.gmd.de 1.0
User-Agent: coco/1.3
From: Christian Zahl <cz@cs.tu-berlin.de>
To: Henning Schulzrinne <schulzrinne@fokus.gmd.de>
Call-Id: 9510021900.AA07734@lion.cs.tu-berlin.de
Referer: ceres.fokus.gmd.de
Expires: Mon, 02 Oct 1995 18:44:11 GMT
Required: audio/pcmu; port=3456; transport=RTP; rate=16000;
            channels=1; pt=97; net=224.2.0.1; ttl=128,
          audio/gsm; port=3456; transport=RTP; rate=8000; channels=1,
          audio/lpc; port=3456; transport=RTP; rate=8000; channels=1
Required: video/h261;port=4134;transport=RTP;rate=128,
          video/nv;port=4136;transport=RTP;rate=250,
          application/x-wb;port=1236
Accept: application/editor;port=3500
Phone: +1 413 555 1212
Location: Paris, Texas
Email: Christian Zahl <cz@cs.tu-berlin.de>
Location: Technical University Berlin;tz=MET;loc=52 32 00 N 13 25 00 E
Priority: urgent
Reach: first
Key: C7 48 90 F4 27 7B A1 CF
Subject: New MUCS error codes
```

Figure 2: Sample MUCS Request

for detecting forwarding loops. The redirect mode is similar in philosophy to the whois++ [20] directory traversal mode. The two modes can be freely mixed within a call.

Note that, after locating a MUCS server, the protocol makes no distinction between mail server hosts and reqular hosts.

Server-side group addresses such as `info@fokus.gmd.de` can be resolved either as *first* or *all*, specified by the Reach header. The *first* mode terminates the call attempts as soon as the first one from the group answers, while *all* contacts all possible candidates and invites them to participate. A reach of type first is appropriate for applications like help lines where the caller is satisfied to talk to any one person from a group, while a reach of type all is useful for setting up multiparty conferences.

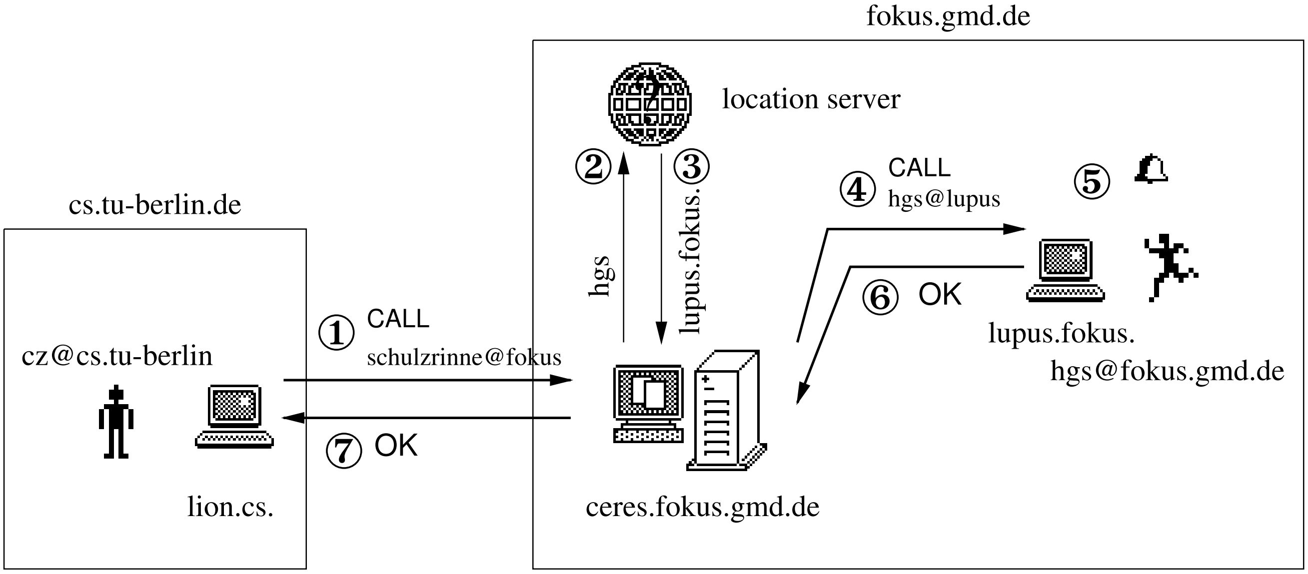

Figure 3: MUCS operation in proxy mode

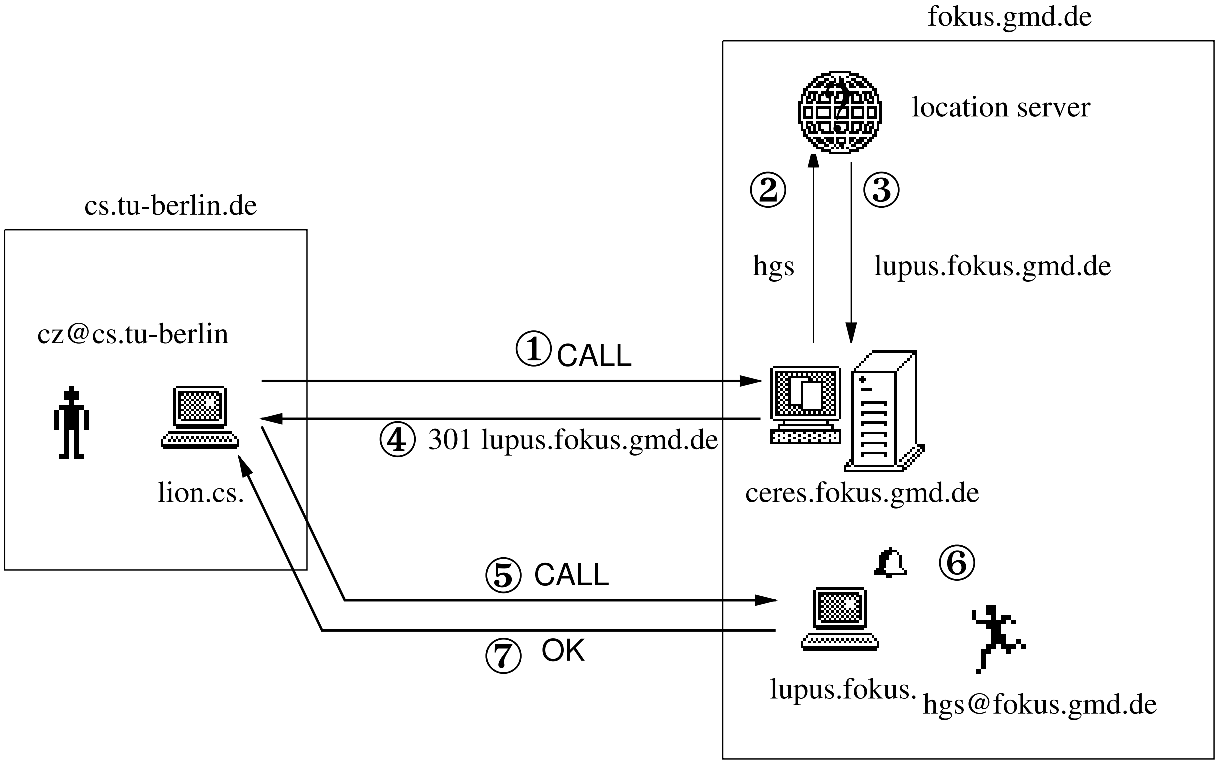

Figure 4: MUCS operation in redirect mode

3.2 Parameter Changes

It is often desirable to add media or change media parameters during a videophone call. The protocol supports this by offering the CHANGE request, together with the Call-Id header. Definitions for all media are sent again as either Accept or Require headers, in the same order as when first establishing the call. Generally, the list of media types should be a subset of those originally listed. It is not a good idea to suddenly require a medium which was optional to begin with.

The media change request can be used to implement a simple centralized media negotiation facility. In the first round, the call initiator contacts each party, gathering the subset of media acceptable to all, the most important conferees, the majority or some other policy. The initiator then distributes this subset of media to all callees that have accepted the call through change requests.

If a server operates in proxy mode, it should cache active call identifiers and their respective user location in stable storage to avoid having to locate the user again. However, should it "forget" a particular call, for example, after a reboot, it can always treat a change request like a newly arriving call and locate the user. If a host has an active user, but no record of the call, it treats it like a newly arriving call.

3.3 Conference Membership Information

Membership in a conference is best described through media participation, with the conference membership as the union of the membership sets for each individual medium[10]. Thus, network outages or members turning off particular media agents can change the conference membership. Also, we do not require that all conference members are invited by MUCS; others may join based on information from, say, a directory service (see Section 6).

When the conference initiator invites participants, complete membership information is not available, and thus cannot be distributed. For multicast media distribution, it appears best to distribute membership information periodically via multicast, using RTCP [17] for real-time media. Maintaining membership information for conferences run over a web of unicast connections or per-site point-to-multipoint connections, say, in ATM or ST-II, is cumbersome without a central conference registry or a distinguished conference initiator. Both of these are undesirable, as they make it difficult to continue conferences after the initiator has left or the central registry is no longer reachable.

Another approach to maintaining a membership list has been explored by the Sticky conference control

[10] Unless there is a "required" medium, it is quite possible that two members of such a conference have no common communication medium.

protocol [21], where transitive closure of reachability is passed around. This approach is also usable here. If the initiator has contacted the initial set of members, it can then issue a change request to all members, containing To fields for all. If the called parties are contacted sequentially, each party can in turn call all the other parties listed, using the same Call-Id.

Because of these complications and since ST-II is rarely used in the Internet context, we currently only support two-party point-to-point and multicast calls.

3.4 Terminating a Call

There is no explicit call termination mechanism within MUCS. If a participant wants to leave a particular media, it uses per-media mechanisms to signal that to the other group members. For RTP, an RTCP Bye packet is sent. For unicast, the sender will receive a port unreachable ICMP message if it continues to send after the other side has closed the socket. All RTP based applications also implement a time-out mechanism that curtails sending to a particular host if that host has not been heard from recently, either via a data or RTCP (control) message. At the network layer, IP multicast requires no additional mechanisms, since data distribution to a particular host will cease automatically when the last group member leaves the multicast group.

The call server on the host where the conferencing applications are actually run (i.e., the final destination of the call request), maintains as a file in /tmp/*Call-Id* a description of the call state. This file will be automatically removed when the system is rebooted and can be used to restart media agents.

3.5 Security

For security, all exchanges can be sent as PGP encapsulated messages, either authenticated or encrypted. Encryption is useful if the identity of the caller or the subject of the call should only be revealed to the callee and not to any of the MUCS proxy servers. Only the request line has to remain unencrypted. PGP keys are also email addresses, so that the re-use of these addresses by MUCS yields another benefit.

Other security mechanisms like the Secure Socket Layer (SSL) [22] or IP-level authentication or encryption could be used. If the CALL request is encrypted, in can also be used to transfer a media session key (the Key header in Fig. 2).

3.6 Interaction with Other Signaling Protocols

The MUCS protocol has to interact with a number of other "signaling" protocols in the Internet. Audio and video applications may use RTP [17] and its associated signaling protocol, RTCP. Each participant periodically multicasts an RTCP packet containing information about itself, such as the user name, real name, physical location, email address, and the like (SDES information), about the amount of data it has sent and reception reports indicating the quality of service for other hosts. The RTCP messages also serve as a connectivity indicator, particularly when hosts send media data only sporadically, as audio might be during a conference. Some of the SDES information can be distributed via the MUCS protocol, avoiding duplicate transmission of the same information with different media. However, given MUCS's use of point-to-point TCP connections, RTCP is a better match for multicast conferences.

The MUCS protocol does not reserve network resources. It is expected that individual applications, when opening network sockets, will also issue resource reservation requests, usually through the resource reservation protocol (RSVP) [23]. For most audio encodings, the necessary information for bandwidth reservation is implied by the media type, for video and other sources, the media description contains the desired bandwidth. The separation of concerns, with MUCS handling invitation and cross-media setup and RSVP responsible for resource reservation, has the advantage that either could be exchanged for a different protocol or omitted. However, the separation into two protocols does introduce an additional failure mode, in that a called party could accept a call, but the resource reservation might fail. A called party could delay accepting a call until the resource reservation succeeds, possibly tailoring its Accept responses accordingly. This has not been implemented. We also anticipate that, if Internet videophone service is to be useful, call failures due to busy or no answer are going to be much more common than those due to lack of network resources, just as in POTS.

RSVP, in turn, communicates with the reservation and signaling protocols of the underlying protocol stack, in particular ATM Q.2931 or SPANS signaling. This interaction is a subject of on-going research and beyond the scope of this paper.

4 Implementation

Large parts of the MUCS architecture have been implemented on a Unix platform. The components and their interaction are indicated in Figure 5. The figure shows only the direct interaction between a caller and a conference daemon, without mail host or forwarding.

The conference daemon (call server) process is started by inetd when a call arrives. The conference controller (call client), called coco, is responsible for originating and answering calls. It is started either by the

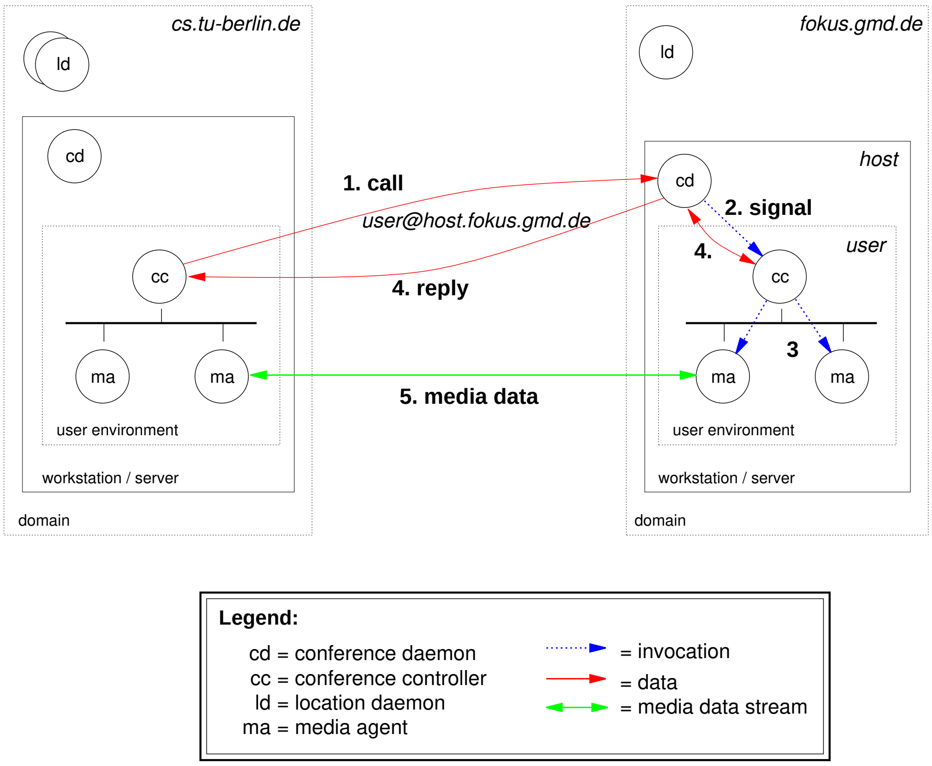

Figure 5: MUCS Components

user or by conference daemon when a call arrives. The conference daemon maps the name part of the video-phone address to a local user name, using the same database employed by the mail transfer agent (MTA). If a call arrives, a user-specified Tcl script is executed in a safe environment, allowing automatic forwarding of calls, similar in spirit to the ideas presented in [24] for interfacing with a traditional telephone or the scripts used for forwarding and filtering electronic mail. If the script has not disposed of the call by forwarding or rejecting it, the conference daemon tries to determine if the called party is logged on the local console. If so, a pop-up window appears asking whether to accept, forward or reject the call. If not, the location daemon is queried for the current location and the call forwarded to that host. If the user is not logged in anywhere, the call is rejected with an appropriate status code. If active badges [7] are in use, the call might be forwarded to a console located in the office visited by the badge owner. (This has not been implemented yet.)

The conference controller uses the local multicast message bus described in [25] to set up and change media configurations.

A sample script illustrating some simple actions is shown in Fig. 6. In the example, calls after 6 pm are forwarded to a home "number", while calls whose subject header contains the name of project, are forwarded to the appropriate person. The `From: *` action invokes a script named after the calling party, if available.

```
Priority: urgent       {ring firebells.au}
From:  my_boss         {ring hail_to_chief.au; ask}
From:  @insurance.com  {reject busy}
From:  *               {$From}
Subject: TOMQAT        {forward deffner}
if {$hour > 18} {
  forward hgs@home -reason "Gone home for the day"
}
```

Figure 6: Sample Call Handling File

Additional forwarding services, such as forward only if the intended receiver accepts the call, or call handover with prior audio contact, can be easily implemented.

If a user cannot be found on the local host, the conference daemon contacts the location daemon and tries to find the user's likely location (Section 2.2) and returns that location to the conference controller (redirect mode). The conference controller can be configured require manual confirmation for redirections within the same domain or only if outside the called domain. If the conference daemon cannot find such a location (say, the user has left for the day), the daemon logs the call and rejects it.

5 Performance Issues

Call volume differs dramatically in different settings, spanning the gamut between private residences attached to an Internet service provider, a research corporation with mostly internal calls to a telephone-sales outfit. A typical local office sees three 3-minute calls in the busy hour per line [26, p. 141], that is, 3 BHCA (busy hour call attempts). A large PBX for 10,000 stations is designed for about 100,000 busy hour calls, i.e., BHCA of 10. Typical dimensioning of outside lines for offices assumes about one outside call per hour and station. Also, ITU recommendation Q.543 lists a BHCA range from 1.2 to 6.8. Thus, even a mail server responsible for handling MUCS needs for several thousand staff members would see a load far less than a moderate-usage web server. A local network serving 200 employees would see a location request at most once every two seconds during the busy hour, which seems manageable even if broadcast were to be used.

Signaling delays are also bounded by telephony expectations. Q.543 (Table 13) specifies, for example, that the delay from end of dialing to start of ringing should have a mean value less than 650 ms, while Q.709 specifies a maximum delay of 1170 ms. This delay limits the number of forwarding operations which can be performed.

6 Related Work

A number of conference control protocols have been described and implemented. Here, we only consider those that do not require a conference server. The Conference Control Channel Protocol (CCCP) [27] allows communication between different components of a conference, either located on a single host or distributed. It assumes that conference members have been notified or invited by other means, one of which may well be MUCS.

The Connection Control Protocol (CCP) [28], partially implemented in the `mmcc` conference controller, supports invitations and negotiation. The protocol is far more complex than MUCS, with numerous states and its own multicast reliability mechanism. Unlike MUCS, it is active during the whole conference. CCP does not support the notion of forwarding, does not interact with SMTP and implements no call handling features. MUCS is also an attempt to show that the functionality actually used for small-group communications can be implemented in a very simple manner.

On the MBONE, seminars, space shuttle launches and other events are announced using the session directory `sd` developed at Lawrence Berkeley Laboratory. It offers a single, non-hierarchical "TV guide" to upcoming events and also allocates multicast addresses. `sd` supports viewer-initiated conferences, with notification of events usually provided through mailing lists or newsgroups. `sd` is part of the light-weight session model [29], where participants communicate principally via IP multicast and maintain "soft" (i.e., fragile and periodically refreshed) state. The MUCS protocol currently operates under that model for the interaction of individual media, but can be used in other environments.

Integration of conferencing and the World-Wide Web has also been pursued. In the Virtual Places approach [30], a subscriber could be met via multiple media at different places (that is, pages) within the web. A very much simplified version can be offered by having a MUCS link in one's home page which simply returns the current videophone address as a URL of method 'mucs'.

7 Conclusion and Future Work

The MUCS protocol allows to establish multimedia conversations across the Internet, offering personal mobility without changes to the existing Internet infrastructure. Together with sufficient bandwidth and resource reservation, it provides another missing component on the way to the multimedia Internet. Users can be reached even if only their own workstation supports the protocol, rather than the site's mail server or dedicated MUCS server. Lifelong personal and trans-national videophone addresses can be offered by any number of service providers, without need for coordination. The call setup protocol can be used with a number of different resource reservation protocols; it works best, however, if the network supports IP multicast and receiver-based resource reservation. The services can be deployed without any changes to the network itself, unlike the substantial, coordinated switch upgrades envisioned for offering UPT services. Since call handling information is only kept on the user's premises, privacy is enhanced and programmed, rather than table-driven call handling can be readily offered. The forwarding service can be offered by competing service providers that are distinct from the bit carriers, making it easy to use several different providers depending on time-of-day, caller, and the like. The MUCS protocol also needs a multicast address allocation service [31].

The re-use of the existing electronic mail address also allows to benefit from the existing infrastructure for email addresses, from address books to mailing lists, PGP keys and aliases.

Interoperation of this MUCS protocol with plain old telephony (POTS), mobile telephony and ISDN remains to be investigated. Telephone numbers in ITU E.123 notation can be used as user names. Together with existing proposals for the integration of email and fax [32] and access to paging services [33, 34], a complete program-controlled communication environment could then be offered.

For personal mobility, the subscriber must have the possibility of securely changing the forwarding characteristics. This can be done remotely with a WWW browser, or an email messaging interface, amongst others.

Despite its limitations when invoking complex functions, the basic telephone set has a human-"computer" interface that even the most techno-illiterate can handle. We are investigating the advantages of attaching a simple telephone set to a workstation, so that a user can accept an Internet phone call in the accustomed manner, by picking up the receiver. Since echo cancellation for speaker phone operation remains difficult and acoustic privacy is often desirable, a phone receiver remains also an attractive audio I/O device.

With the advent of high-speed IP connectivity provided by ISDN or cable TV, Internet telephony may offer advanced services far sooner (and possibly cheaper) than traditional telephone operators.

8 Acknowledgements

Discussions about PBX and central office sizing with the GMD telecommunications office and M. R. Lundberg were very helpful.

References

[1] H. Schulzrinne, "Internet services: from electronic mail to real-time multimedia," in *Proc. of KIVS (Kommunikation in Verteilten Systemen)* (K. Franke, U. Hübner, and W. Kalfa, eds.), Informatik aktuell, (Chemnitz, Germany), pp. 21–34, Gesellschaft für Informatik, Springer Verlag, Feb. 1995.

[2] H. Eriksson, "MBONE: The multicast backbone," *Communications ACM*, vol. 37, pp. 54–60, Aug. 1994.

[3] C. E. Perkins and P. Bhagwat, "A mobile networking system based on the internet protocol," *IEEE Personal Communications*, vol. 1, pp. 32–41, First Quarter 1994.

[4] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended LANs," *ACM Transactions on Computer Systems*, vol. 8, pp. 85–110, May 1990.

[5] R. Pandya, "Emerging mobile and personal communication systems," *IEEE Communications Magazine*, vol. 33, pp. 44–52, June 1995.

[6] D. C. Lynch and M. T. Rose, *Internet system handbook*. Reading, Massachusetts: Addison-Wesley, 1993.

[7] A. Hopper, "Communication at the desktop," *Computer Networks and ISDN Systems*, pp. 1253–1265, July 1994.

[8] D. Crocker, "Standard for the format of ARPA internet text messages," Tech. Rep. STD 11, RFC 822, Internet Engineering Task Force, Aug. 1982.

[9] B. Kantor and P. Lapsley, "Network news transfer protocol: A proposed standard for the stream-based transmission of news," Tech. Rep. RFC 977, Internet Engineering Task Force, Feb. 1986.

[10] T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext transfer protocol – http/1.0," Internet Draft, Internet Engineering Task Force, Aug. 1995. Work in progress.

[11] N. Freed and A. Cargille, "SMTP service extension for command pipelining," Tech. Rep. RFC 1854, Internet Engineering Task Force, Oct. 1995.

[12] J. K. Ousterhout, *Tcl and the Tk Toolkit*. Reading, Massachusetts: Addison-Wesley, 1994.

[13] C. Partridge, "Mail routing and the domain system," Tech. Rep. STD 14, RFC 974, Internet Engineering Task Force, Jan. 1986.

[14] A. Gulbrandsen and P. Vixie, "A DNS RR for specifying the location of services," Internet Draft, Internet Engineering Task Force, Mar. 1995. Work in progress.

[15] R. Braden, "T/TCP – TCP extensions for transactions functional specification," Tech. Rep. RFC 1644, Internet Engineering Task Force, July 1994.

[16] M. Handley and V. Jacobson, "SDP: Session description protocol," Internet Draft, Internet Engineering Task Force, Aug. 1995. Work in progress.

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications." Internet draft (work-in-progress) *draft-ietf-avt-rtp-*.txt*, Mar. 1995.

[18] S. Shenker, A. Weinrib, and E. Schooler, "Managing shared ephemeral teleconferencing state: Policy and mechanism," Internet Draft, Internet Engineering Task Force, July 1995. Work in progress.

[19] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP," in *First International Workshop on High Speed Networks and Open Distributed Platforms*, (St. Petersburg, Russia), June 1995.

[20] P. Faltstrom, R. Schoultz, and C. Weider, "How to interact with a Whois++ mesh," Internet Draft, Internet Engineering Task Force, Mar. 1995. Work in progress.

[21] C. Elliott, "A 'sticky' conference control protocol," *Internetworking: Research and Experience*, vol. 5, pp. 97–119, 1994.

[22] K. Hickman and T. Elgamal, "The SSL protocol," Internet Draft, Internet Engineering Task Force, June 1995. Work in progress.

[23] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "Rsvp: a new resource ReSerVation protocol," *IEEE Network*, vol. 7, pp. 8–18, Sept. 1993.

[24] S. A. Uhler, "PhoneStation, moving the telephone into the virtual desktop," in *Proc. of Usenix Winter Conference*, (San Diego, California), pp. 131–140, Jan. 1993.

[25] H. Schulzrinne, "Dynamic configuration of conferencing applications using pattern-matching multicast," in *Proc. International Workshop on Network and Operating System Support for for Digital Audio and Video (NOSSDAV)*, (Durham, New Hampshire), pp. 231–242, Apr. 1995.

[26] B. E. Briley, *Introduction to Telephone Switching*. London: Addison-Wesley, 1983.

[27] M. Handley, I. Wakeman, and J. Crowcroft, "The conference control protocol (cccp): a scalable base for building conference control applications," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (Cambridge, Massachusetts), pp. –, Sept. 1995.

[28] E. M. Schooler, "The connection control protocol: Specification (version 1.1)," technical report, USC/Information Sciences Institute, Marina del Ray, California, Jan. 1992.

[29] V. Jacobson, S. McCanne, and S. Floyd, "A conferencing architecture for light-weight sessions," Nov. 1993. MICE seminar series (transparencies).

[30] E. Shapiro, "Virtual places – a foundation for human interaction," in *Proc. of the Second World Wide Web Conference'94*, (Chicago, Illinois), Oct. 1994.

[31] S. Pejhan, A. Eleftheriadis, and D. Anastassiou, "Distributed multicast address management in the global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1445–1456, Oct. 1995.

[32] C. Malamud and M. Rose, "Principles of operation for the TPC.INT subdomain: Remote printing – technical procedures," Tech. Rep. RFC 1528, Internet Engineering Task Force, Oct. 1993.

[33] M. Rose, "Principles of operation for the TPC.INT subdomain: Radio paging – technical procedures," Tech. Rep. RFC 1703, Internet Engineering Task Force, Oct. 1994.

[34] A. Gwinn, "Simple network paging protocol - version 2," Tech. Rep. RFC 1645, Internet Engineering Task Force, July 1994.