The Multi-Media Telephone:

Directory service and session control for multi-media communications

Andrew Findlay

Computer Centre
Brunel University
Uxbridge
GB

Andrew.Findlay@brunel.ac.uk

January 1996

ABSTRACT

Multi-media communication tools now exist which have support for group communication using multicast protocols. The same tools can be used in unicast mode for one-to-one communication.

The focus of development so far has been on group-working and conference support. As a result, there is a session directory tool that allows users to advertise conferences and to join the ones that interest them. On the other hand, the support for setting up one-to-one sessions and closed-group conferences is rather limited.

A new set of session-control facilities is needed to create a 'multi-media telephone' from the components now available. This paper lists some requirements and proposes mechanisms to address them.

1. Requirements

It must be easy to set up a point-to-point call. It should be enough to give the name of the person you want to talk to without having to know the machine where they are currently working.

The system must scale to a very large user community: if it is to rival the global telephone network it must cope with over 100 million users.

People must be able to move about without their 'address' changing. In telephone parlance, this is closer to 'roving' than to 'personal numbering'. It is reasonable to expect someone's address to change if they move to a new job, but not if they are simply working in a different place from where they were a week ago. The distinction is important, as personal names are far from unique and individuals are commonly identified by reference to where they live or where they work.

The system must support the use of separate tools for each medium as well as integrated ones. The user expects to see a well-integrated environment, but it is often better to implement this as a set of co-operating tools rather than a single monolithic application.

Security services will be required, including end-to-end privacy, proof-of-identity, and options for anonymous calling. It should also be possible for either party to hide their actual location from the other.

All parties will require call-status information, including the equivalent of ‘ringing’ ‘engaged’ ‘unobtainable’ ‘hangup’. There will be more possible states than with a conventional telephone, as each medium involved in the call could be handled separately: a call may be answered for voice but still ‘ringing’ for video for example.

There must be support for conferences and meetings, including non-public meetings, broadcasts (where only one endpoint may transmit), meetings with observers (certain defined parties may transmit but others can only receive).

It must be possible to add and remove parties at any time: a session that starts as a point-to-point call might expand into a conference, then divide into several smaller groups, some of which might become point-to-point calls again before disconnecting. Some conferences will require all parties to be nominated by a ‘chairman’ where others will wish to accept calls from anyone, possibly vetting each new caller before allowing them to join the main conference.

Answering services must be supported (though the straight answerphone-replacement might be discouraged in favour of carrying voice in e-mail)

The system must support ‘organisational’ endpoints as well as personal ones. This would include ‘switchboard’ functions, call distribution to helpdesks, call transfer, and various actions to be taken if the callee is busy or does not reply within a certain amount of time.

A particularly difficult service to support will be the ‘nearest available service centre’ routing used by emergency services and other large organisations that operate from many locations, each serving a defined geographical area. As there is not likely to be much correlation between network topology and geography the conventional calling-address routing will not work. It will be necessary for each end-point to have some notion of its own location and to use this information to reach the appropriate service centre. Note that this facility may conflict with the confidentiality-of-location service mentioned above, so the user may need to be actively involved in any exchanges involving such data.

Although the multi-media-telephone is proposed as an IP-based service, the requirements for interworking with other transport media must be addressed. Multi-media tools exist that use ‘raw ATM’ ‘raw ISDN’ and, of course, POTS: the Plain Old Telephone Service.

2. Service Architecture

The main functions to be performed are:

Directory Service
Location Service
Session Control
Media Handling

Each function requires user agents, and most require service agents as well (Figure 1). Each function will use its own protocol, and communication between functions will be minimised to preserve modularity.

3. Directory Service

At the simplest level, the Directory Service performs the function of an electronic phone book. It must be a worldwide integrated service designed to minimise the knowledge required by the user and user-agent. It must provide a globally-unique name for each user, and be able to store a variety of information relating to users and other entities. Every significant entity in the multi-media telephone system will be identified by it’s Directory Name (DN). The most appropriate service model is X.500[ISO88a, ISO93a] which is designed to scale to hundreds of millions of entries but only requires the user to contact a single Directory Service Agent (DSA) to obtain any information on any other user.

The Directory Service will do more than just name-to-address translation: a subscriber’s entry may well contain pointers to message services and could even hold further information to allow a potential caller to check that they really have found the right person to call. The directory entry will also contain credentials and other security information necessary to verify and protect communications.

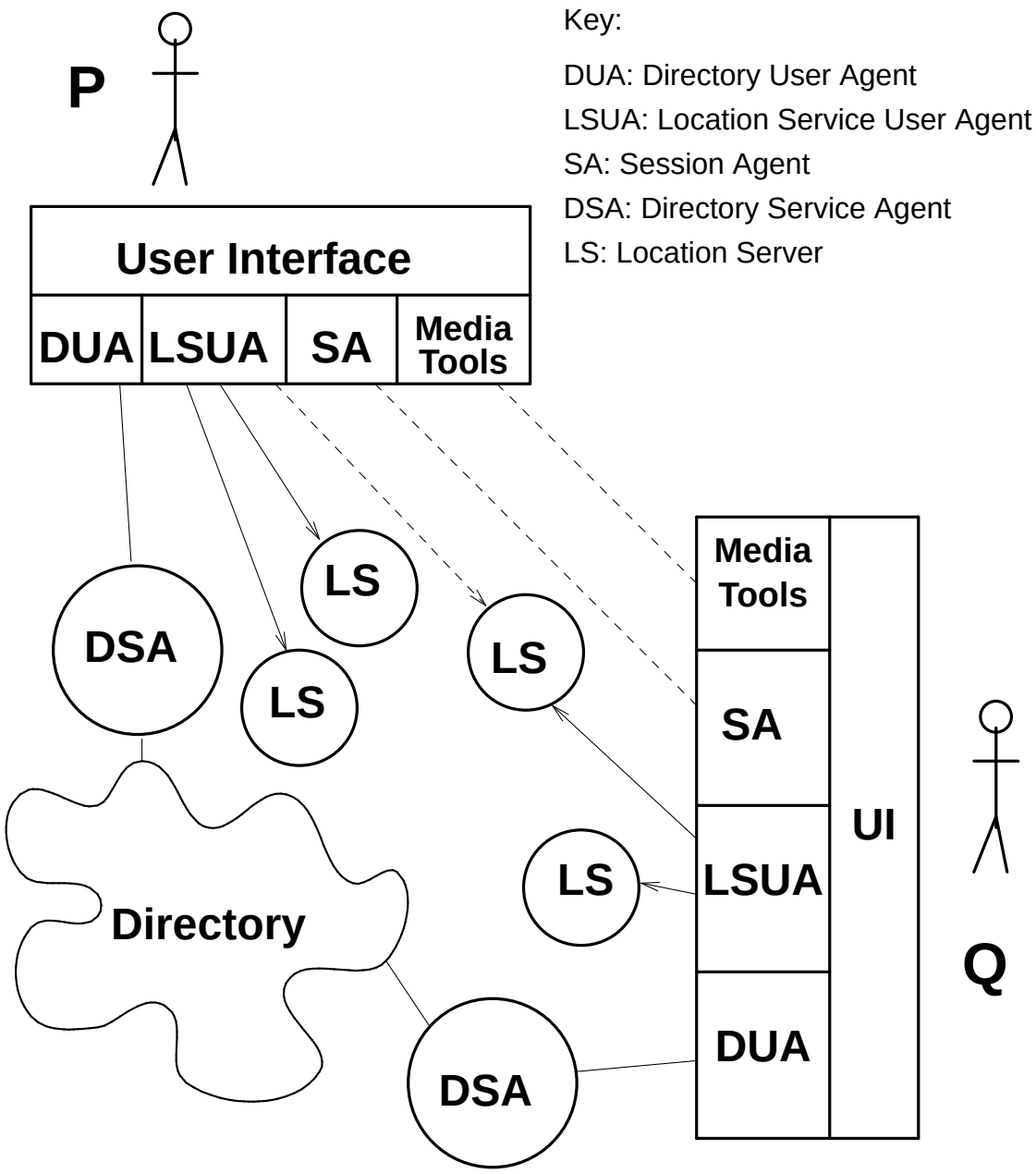

Figure 1: Service architecture

Figure 1: Service Architecture

Directory Services are designed to hold data that does not change rapidly: this allows replicated servers and large caches to work efficiently. It would be reasonable for a user's directory agent to maintain a private addressbook of data extracted from the directory and other sources. The data in the addressbook could be used to avoid repeating directory lookups for every call, though it should be periodically verified.

Each user interacts with the directory service though a Directory User Agent (DUA) which normally communicates with a single Directory Service Agent (DSA).

4. Location Service

Users are likely to be mobile. When they move from place to place they may take their communication agents (in a portable device) or they may use public agents (public 'phones' or guest accounts on other networks). In either case it is likely that a user's network address will change from time to time. For some users there could be several changes of address each day. It is unreasonable to make such frequent updates to a global directory service, so a separate Location Service is proposed.

The location service has a single job: it maintains a validated mapping from subscriber DNs to current session agent addresses.

The Location Service is much simpler than the Directory Service. Location Servers do not need to communicate with each other: the only requirement is that all users' location service agents can communicate with all Location Servers that they subscribe to, and with at least one Location Server that holds data for the intended destination of the next call. There can be as many location servers as necessary, and users might subscribe to more than one for resilience.

Whenever a session agent starts up or changes address, it will register its new address with all location servers listed in the user's directory entry. At shutdown, the session agent may choose to de-register the address or to register the address of an answering service. If the session agent becomes disconnected without informing the location service, callers will be told that the user is currently unreachable.

Each user interacts with the location service though a Location Service User Agent (LSUA) which may communicate with a any number of Location Servers (LSs).

5. Session Agent

Each user will need a session agent to make and receive calls, accept or reject them, and start and stop the communication channels. This is analogous to the telephone instrument itself. The session agent works closely with the various media tools, passing instructions and parameters to them when session state changes occur.

At startup, the session agent binds to an available unicast address and registers this with the location service as described above. Any incoming calls from other session agents are verified and presented to the user for answering: this is analogous to the phone ringing.

Each user will normally have a single Session Agent (SA). Session Agents communicate directly with each other: there is no Session Server.

6. Security services

It is important that communications are properly authenticated. This is particularly true of the messages used to register with a location server: if these were not protected it would be easy for an imposter to divert any user's calls to any desired location.

All users will have cryptographic credentials, which will be stored in their directory entries. The mechanisms defined for Privacy Enhanced Mail (PEM) are appropriate here.[Lin93a, Ken93a, Bal93a, Kal93a]

The credentials will be used to sign messages between agents, and also to implement a public-key cryptosystem for protecting session-key exchanges and other sensitive traffic.

It is intended that *all* communications proposed by this paper should be signed and protected from eavesdropping by cryptographic means. The only messages that need not be signed are those requesting information from directory and location services. Even there, it is preferable to sign the messages as the services concerned may be programmed to only give out certain information to certain individuals. (Such selective service would be appropriate to implement various forms of ex-directory facilities)

7. Point to Point calls

The process of making a call from user P to user Q is as follows (see Figure 1):

(1) P's DUA looks up Q in the directory. The result is Q's Directory Name (DN) and credentials, and also the DNs of one or more location services that Q subscribes to. All of this information may be cached in P's addressbook.

(2) P's DUA looks up one of Q's location services in the directory, to get its addresses and security information. This information may also be cached in the addressbook.

(3) P's LSUA then calls the location service and presents Q's DN. The location service replies with the address of Q's session agent. This information was signed by Q when it was registered, so P's LSUA can easily verify its authenticity.

(4) P's session agent contacts Q's session agent and presents the call request. This will include P's DN and the current address of P's session agent, and may also include the subject of the call and the relative urgency. The call request will be signed by P.

(5) Q's session agent may handle the request automatically (I'm always out to market research companies and salesmen) or may alert Q.

(6) If Q accepts the call, the two session agents agree on communication parameters and call up appropriate media tools to handle the session. The session agents remain connected until the end of the call to handle function such as 'hold', 'transfer', 'add participant', and 'hangup'.

It should be noted that the first three steps can be omitted if P and Q have communicated before and have cached the results of the lookups in their local addressbooks. In this case, a failure to make contact should force one or more of the omitted steps to be performed in order to verify and update the cached data.

Any call that involves more than two end-points should switch to multicast communication, or might use multicast from the start if it was known in advance that this was appropriate. When a participant leaves such a multi-party call they can be positively excluded from further participation if necessary by changing the session keys in use.

The transition from point-to-point operation to multicast must be handled carefully to avoid breaks in communication:

(1) The user wishing to bring another party into the call locates the session agent of the new party using the mechanisms already defined, and sends a 'call' request.

(2) While this is going on, the two original session agents agree a multicast address to be used for the session and start exchanging multicast session announcements using that address. The media tools are instructed to accept data on the new multicast address as well as the original unicast address.

(3) As soon as both session agents have verified the multicast link they instruct the media tools to switch to multicast transmission. The timestamps in the media flows should prevent 'glitches', though if the latency of the multicast route is greater than that of the unicast route there will still be a break in the data stream. The effect can be minimised by switching at a 'natural break' in the audio stream.

(4) If the new party decides to answer the call, their session agent obtains the communication parameters from the calling agent and starts the appropriate media tools directly in multicast mode.

An alternative situation would be for a new participant to call *in* to a conference. In this case there might be a unicast discussion with a conference controller before the caller is brought into the multicast group: this would be appropriate for groups where the participant list is not known before the start of the call.

For pre-arranged meetings, the multicast addresses and other parameters could be determined in advance and included with the notice of the meeting. Joining such a meeting would be very simple, with no negotiation or service location steps needed. The same process would be used for 'broadcast' events.

8. Proxies

The explanations above relate to simple point-to-point communications, which are appropriate for calls to individual named people. Organisations have extra requirements: they have 'functional' contact points such as sales offices and helpdesks, they need automatic call routing of various forms, location independence, and many other 'added value' features. Organisations are often insulated from the global network by firewalls (switchboard operators perform a similar function to Internet Firewalls in many companies!)

Although many of these functions could be handled by the session agents and some clever location services, the extra level of indirection offered by proxy agents will be of great benefit to many organisations. Outside callers will connect to the proxy, which then handles all communication within the organisation. Each part of the service will need its own set of proxies, which will be controlled from 'inside' the organisation that they serve.

An organisation or department using a proxy to insulate it from the 'outside world' would run its own Directory and Location services on the 'inside'. All internal users would register with these internal services to support local communication. Only those people who wish to receive calls directly from outside would instruct the proxy to register them with an externally-visible location server.

9. Group Communications

Organising meetings is rather different from the current advertised conference service implemented by SD[Jac93a] for several reasons:

> Meetings are usually private to the invited attendees. The existence of the meeting is often confidential information as well.
>
> Although there are only a few open conferences running at any one time at present, by the time this is scaled up to be a global communication service there will be many thousands. Some will have millions of participants (think of a public TV channel as a one-way conference....) while others will have just two.

The usual way of organising meetings is by phone or by memo, stating a time and place. There are beginning to be distributed scheduling systems: this aspect is not discussed here, we are interested in how the meeting details are communicated to the invited participants. The obvious medium is e-mail, so a MIME[Bor93a] body-part format to carry session details would be an obvious thing to use. The details to be carried would include:

> Description of the meeting
> Communication parameters
> Date and time, expected duration
> Encryption keys and credentials

A clever mail agent would be able to put this information into a diary, and the diary in turn could alert the session agent at the appropriate time.

Public meetings and broadcasts can be advertised by similar means, with the message being sent on any appropriate electronic medium such as Usenet News or the Web. The process of 'joining' such a public channel would be slightly different from joining a closed meeting: the announcement would normally contain enough information to avoid having to contact directory and location services, and the session agent would start the media tools in a receive-only mode. This mechanism could even could be extended to handle some forms of pay-TV.

10. Existing work

Many groups have done important work in the Multi-Media communications area. The most relevant projects to build on include:

> MICE (UCL)
> CAR (UCL)
> MMCC (University of Southern California)

Of these, MMCC[Sch94a] has come closest to the 'multi-media telephone' application and MICE (and its forerunners) has the most comprehensive conference-control proposals.[Han95a] The present proposal builds on these mainly in the areas of scaling to very large user communities and provision of integrated directory services. It duplicates some of the work currently being undertaken in the IETF MMUSIC working group on conference invitation protocols, though the author's view is that the architecture proposed here is much cleaner than the current SCIP draft[Sch96a] and has wider applicability. In particular, SCIP does not address directory service integration and it depends on features of the current HTTP and SMTP protocols to

achieve a location service and session control protocol.

There have recently been a number of Internet Telephone products announced to the public. Some are commercial and some are in the public domain. Most run on MS-DOS PCs, though some have versions for other types of hardware. Quarterdeck's WebTalk is getting good reviews for usability and sound quality, but it's centralised addressbook service comes in for some criticism. Similarly, WebPhone (which actually looks like a mobile phone handset on-screen) has a centralised address service but other products have little provision for locating other users at all. Some products (e.g. Silversoft's SoftFone) appear to have a form of location service, recognising that many IP service providers allocate addresses dynamically to subscribers at connect time. There do not appear to be any products at present with scalable directory and location services.

At the higher-cost end of the market there are several ISDN-based video conferencing packages such as Intel's ProShare. These tend to use local addressbooks to hold the ISDN phone numbers, again limiting the ease of use in an open user community.

11. Shortcuts

To get a usable service going reasonably fast for experimental purposes it would be reasonable to relax some of the requirements:

By relaxing the security requirement it would be possible to simplify things a lot: there would be no need for PEM certificates and the facilities needed to generate them.

The session agent and location service could be run without the directory service, though users would need to pass arcane identifiers around. This should be regarded as a transient stage, as a high-quality service will certainly need a directory service.

It may be possible to use WWW tools as front-ends for the calling side of the session agent.

12. Conclusions

A framework appropriate to a global multi-purpose multi-media communication service is proposed. Although described as a multi-media telephone, the system will also handle group communications ranging in scope from a small meeting up to an international television channel.

Several distinct functions are identified: media tools, session agent, location service, and directory service. It is proposed to implement each function with a separate dedicated protocol. The requirements of each function have been outlined, and suggestions made for implementations.

References

Bal93a.

D Balenson, ''Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers,'' *RFC1423* (February 1993).

Bor93a.

N Borenstein and N Freed, ''MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies,'' *RFC1521* (September 1993).

Han95a.

Mark Handley and Ian Wakemen, ''CCCP: Conference Control Channel Protocol,'' *ftp://cs.ucl.ac.uk/mice/publications/cccp.ps.gz,* London (August 1995).

ISO88a.

ISO/CCITT, ''Recommendation X.500: The Directory - Overview of Concepts, Models and Services,'' Geneva (March 1988). ISO 9594 is technically aligned with X.500.

ISO93a.

ISO/ITU-T, ''X.500(1993) The Directory,'' Geneva (1993).

Jac93a.

Van Jacobson, ''Using the LBL "Session Directory" (sd),'' *SD Documentation,* LBL (March 1993).

Kal93a.
B Kalisky, ‘‘Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services,’’ *RFC1424* (February 1993).

Ken93a.
S Kent, ‘‘Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management,’’ *RFC1422* (February 1993).

Lin93a.
J Linn, ‘‘Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures,’’ *RFC1421* (February 1993).

Sch94a.
Eve Schooler and Joe Touch, ‘‘Multi-Media Conference Control (MMCC),’’ *MMCC program documentation* (September 1994).

Sch96a.
Schulzrinne, H, ‘‘Simple Conference Invitation Protocol,’’ *Work in progress* (Feb 1996).