Secure Inclusion of Phones into Online E-Meetings

Peter Parnes
Luleå University of Technology
Department of Computer Science and Electrical Engineering
Media Technology Division
971 87 Luleå, Sweden.

Peter.Parnes@sm.luth.se

April 18, 2003

Abstract

Online Internet based e-meetings for synchronous communication is becoming more and more common and the need for secure communication is a strong requirement from both corporate and private users. At the same time not all users can always be available via the Internet and as phones (fixed and mobile) are still the dominant communication method there is a need for general inclusion of phones into online e-meetings, without compromising the security of the online session. This paper presents an architecture supporting three different scenarios for including normal phones into e-meetings, and at the same time keeping the security of the session intact. It is also shows how an e-meeting portal is used for simple inclusion of phones into e-meetings even if the inviting client is behind a NAT gateway or a firewall.

Keywords: e-meetings, MBone, multicast, synchronous communication, SIP, IP-telephony

1 Introduction

The need for people to meet over the Internet is becoming pervasive. Meeting scenarios include everything from scheduled meetings for project groups and net-based learning to 24-hour connected e-corridors for shared group awareness and an increased sense of presence. The latter allows users to exchange real-time media via a common group area were all connected users can see the transmitted video and as the number of users increase the need for scalable distribution increases. As a reference, it can be mentioned that the author is daily part of 3-5 different e-meetings or e-corridors and is constantly receiving between 0.5 and 2 Mbps of real-time video data distributed over 10-50 simultaneously sending users.

One such system is the Marratech Pro[1], which allows users to interact synchronously via audio, video, text based chat, shared whiteboard and shared WWW browsing. The client supports both data distribution using IP-multicast [1] for direct peer-to-peer communication and in the case that IP-multicast is not available the Marratech E-Meeting Portal is used as a data distribution server. Figure 1 shows the application user interface for 4 simultaneous e-corridors.

[1] <URL:http://www.marratech.com/>

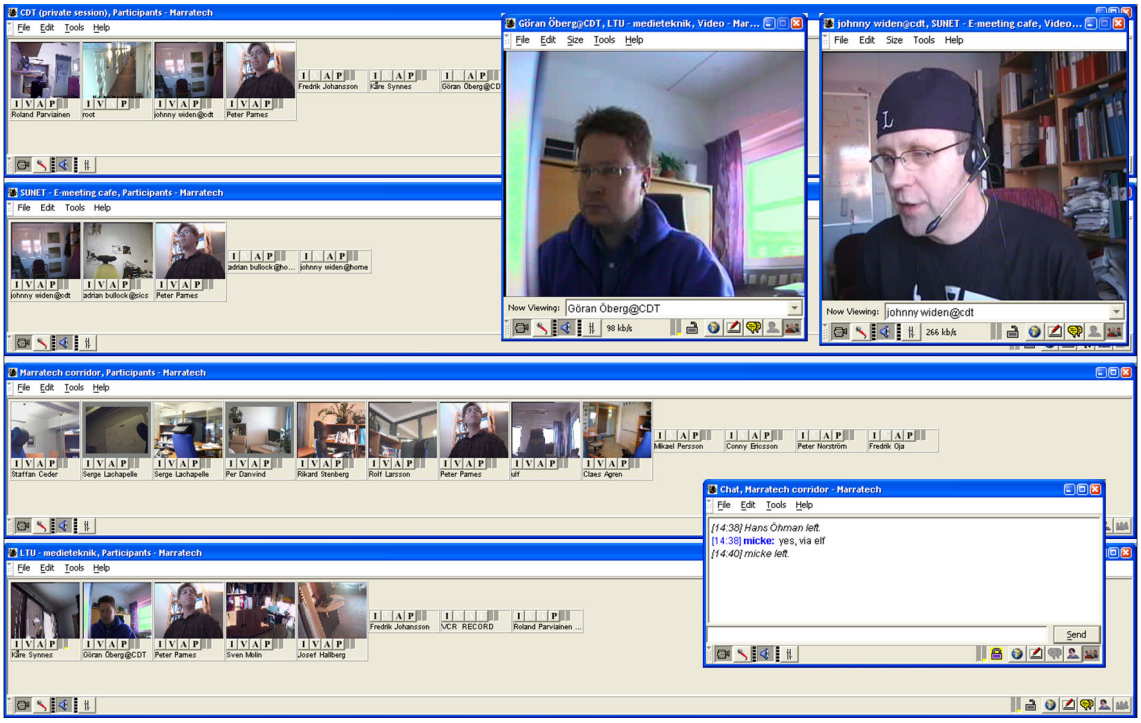

Figure 1: Screen shot of 4 simultaneous e-corridors

Something going hand in hand with on-line meetings is the user requirement for secure communication, where all exchanged media data is encrypted, including the real-time audio and video. It is here obviously attractive that in the case a portal is part of the e-meeting, it should not have to be able to decrypt the media, as the portal might be hosted by a third party (e.g. an ISP or a computer support department) not trusted by the members of the online group. In general, it is always advisable to share a common secret with as few other parties as possible, if it should remain secret. Symmetric encryption is here used due to a private/public encryption scheme does not scale to a large number of users when exchanging high bandwidth real-time media.

All users do not always have access to a local computer to allow for interaction with e-meeting groups and thus it is attractive to be able to join an e-meeting via a normal phone (wired or wireless). This inclusion of phones should be done without compromising the security of the session.

This paper presents an architecture for secure inclusion of phones into online meetings using three different scenarios, where each scenario has certain benefits over the other two. The architecture is primarily designed around the assumption that an existing member of the e-meeting invites phone users into the session, i.e. the phone will get a call. The opposite where the phone user calls directly into the meeting is left for further work and its current status is further discussed in section 5.2.

The rest of this paper is organized with section 1.1 discussing related work and continues with an overview of the proposed system in section 2. Section 3 presents the architecture and its implementation, while section 4 presents the evaluation of the work. The paper is concluded in section 5 together with a discussion about future work.

1.1 Related Work

Usage of IP based online e-meetings have grown over the last 15 years and the first usable applications, such as VIC [2] and VAT [3] were all part of the MBone [4] suite of tools. The authors own work in this area lead to the mStar suite [5, 6] that in turn lead to the creation of the Swedish company Marratech AB. The research around the MBone suite has led to a number of IETF and ITU standards recommendations for IP based online communications, but unfortunately neither the MBone suite nor the ITU H.323 [7] provide a full environment for supporting scalable group communications and a number of incompatible systems have been created over the years. The audio and video parts though are compatible and [8] presents one solution for merging these two conferencing domains using the Session Initiation Protocol [9].

2 Overview

This section presents three different alternatives for solving the problem of including phones into an e-meeting.

First of all, an interface component between IP networks and the phone system has to be included. This can be easily be solved by a dedicated computer with a phone interface (i.e. modem), but that only gives a single phone line and is not a production quality solution.

A more robust solution is to use a dedicated phone gateway that can be accessed via a standardized protocol, such as H.323 or the Session Initiation Protocol, SIP. The latter is a text based open protocol and thus selected for this system. A further advantage of using SIP is that several Internet phone operators exist today that sell IP telephone services based on SIP and it has been the goal of the work presented in this paper to be inter-operable with these Internet phone operators.

2.1 Key Management in E-Meetings

A basic requirement for secure e-meetings is that the actual exchange of real-time media is secure and this is handled via the Secure Real-time Transport Protocol, SRTP [10] which is an extension of the Real-time Transport Protocol, RTP [11]. The shared knowledge (i.e. the key) that is used for encryption can be exchanged in a number of ways. It can be stored in the e-meeting description file, provided by the user at startup or it can be included via an external reference using a URI. The last is the method that is both secure and convenient as the key can be updated regularly[2] and the user does not need to store the key in his/her mind or noted locally. It also means that the key is not shared with the administrator of the e-meeting portal, even if it provides the session information. The remote key should of course be protected by e.g. a normal secure WWW protection scheme.

2.2 Integrating Phone Audio Media into E-Meetings

The phone-gateways are normally constrained to only support one or a few audio encodings as well as not being able to mix several audio streams or even to receive more than audio stream from the IP network. In an e-meeting, audio from each participant is sent in its own media stream to both allow for peer-to-peer communication (i.e. no central unit that mixes all active audio streams) and to allow the user to mute individual audio senders. This leads to the requirement that the e-meeting to SIP gateway has to support both optional two-way transcoding of audio, mixing

[2] If the key is changed it will only take effect after the client is restarted but that is an implementation issue.

of audio streams, and to maintain the security it also has to handle the encryption and decryption of the audio media.

The integration can be done by the following approaches:

A *Client only:* Let the client take care of all the security coding, audio mixing and audio transcoding. The advantage is that no portal is required. The disadvantage is that any computer glitches (such as running an application that takes a lot of CPU, e.g. starting a large word processor application) will affect the audio quality, not only to the local user but also to the rest of the group if audio is actively sent between the phone and the group. Degradable audio for the local user is usually accepted as the user is the one that causes the degradation. The handling of the security information (encryption key) is all done locally.

B *Portal with shared secret:* Move all the audio mixing and audio transcoding to the portal and share the e-meeting encryption key with the portal. The advantage here is that inclusion of the phone into the e-meeting is not dependent on a client host and its CPU as mentioned in approach A. The disadvantage is that the portal has to know the shared secret and once it has the key it will be able to decode the audio even after the call is terminated.

C *Portal without shared secret:* Same as approach B but one client does the decryption, re-encrypts the audio and sends it to the portal for further handling. Here the one client and the portal exchange a common secret to be used for encrypting the media between the portal and the one client. The resulting difference between this and approach B is that the portal only gets access to the audio media, and only during the call is active.

A special case which somewhat unifies approaches B and C, is when different keys are used for each media in a session. This is fully possible, but makes it more cumbersome to set up and potentially more cumbersome for the user. Using different keys is not the common case in today's e-meeting usage. Also, obviously if no encryption is used in the session then there is no shared secret problem at all.

No commercial Phone SIP gateway service allows clients to connect without providing correct credentials and this has to be included in the architecture. The transport of the credentials (i.e. authentication) is handled by SIP.

These different alternatives are further discussed in conjunction with the proposed architecture in section 3 and further evaluated in section 4.

2.3 Handling of NAT and Firewalls

Network Address Translation, NAT gateways and firewalls are very common in today's Internet. If a client is behind a NAT gateway then a Portal can be used to participate in an e-meeting and the only thing required is that the NAT gateway and firewall is setup in so called "allow return" mode, meaning that if traffic has first been sent out through the gateway/firewall then traffic can come back through the gateway/firewall on the same IP port. Allow return is usually the default mode on gateways/firewalls making it very easy to setup an e-meeting.

The authorization process between the client and the e-meeting portal is a two step process. First a secure TCP connection is established using SSL over TCP (i.e. normal secure WWW interaction) and a shared secret as well as public identifier is created by the portal and shared with the client. Session network information, described using the Session Description Protocol, SDP [12] is also passed from the portal to the client including which ports the portal is listening on for each

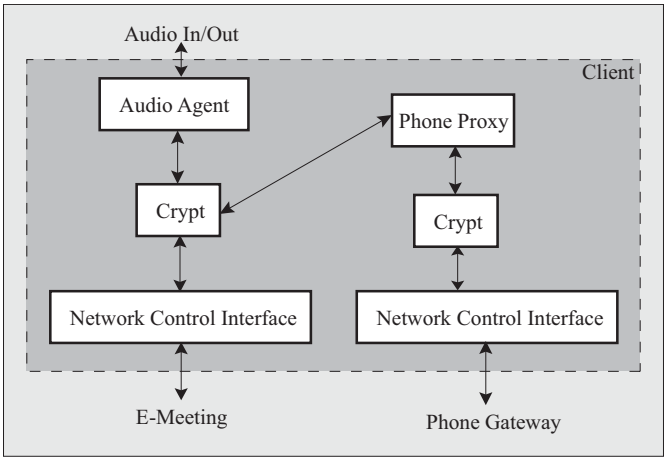

Figure 2: Internal architecture of approach A.

media. The client, then for each media sends a UDP packet to the corresponding port on the portal with the public identifier. This maps the port on the client side (i.e. the outgoing port on the NAT/firewall gateway) to the client, but as this can easily be compromised a challenge/response transaction is initiated based on the shared secret. If the challenge/response transaction was successful the client becomes a full member of the e-meeting. The same port mapping authorization scheme is utilized even if no NAT/firewall gateway is used.

3 Architecture

This section goes further into depth by discussing the architecture of the proposed system.

3.1 A: Client Only

The client only solution does not involve the portal in the process of including the phone into the e-meeting and all SIP messaging is handled directly by the client.

Audio to and from the phone gateway is handled by a *phone proxy* software component embedded into the application and call interaction with the phone gateway is handled by the *SIP agent*, acting as a SIP User Agent Client. The process of inviting a phone into an on-going e-meeting is as follows (simplified):

1. The user opens a user interface dialog and enters a phone number (directly or via a phone book).
2. The SIP agent sends an INVITE message to the phone SIP proxy.
3. The phone SIP proxy answers with an authorization required answer.
4. The SIP agent retrieves the credentials needed (from local storage or by asking the user) and resends the INVITE message to the SIP proxy.
5. The SIP proxy then establishes a connection to the phone itself and sends an acknowledge message to the SIP agent.

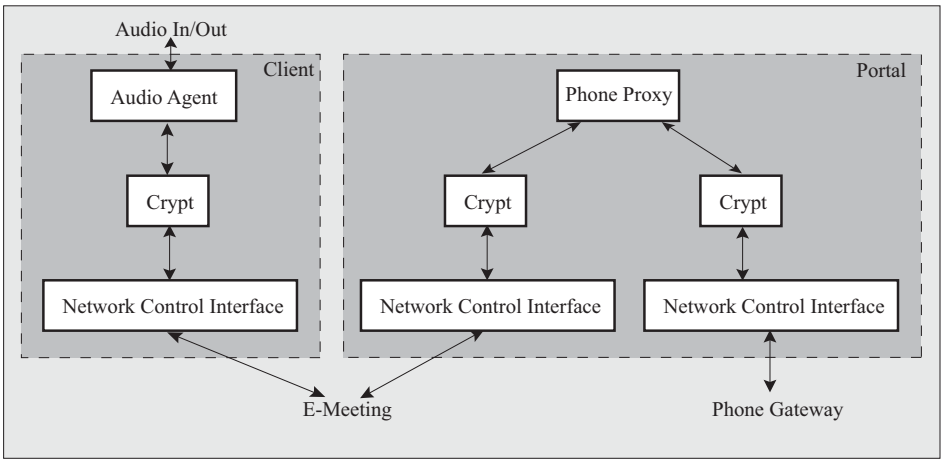

Figure 3: Internal architecture of approach B.

6. The application then sets up the phone proxy that handles en-/decryption, transcoding and mixing of the audio.

7. Audio data can now be exchanged between the phone and the e-meeting.

8. At the end of the conversation the call is terminated by either the phone user (the SIP proxy sends a BYE message to the client) or by the local client user (the SIP agent sends a BYE message to the SIP proxy).

This process is very similar to how a normal SIP based IP telephone call is set up. Note, that if a NAT gateway or firewall (see section 2.3) is present between the client and the phone SIP proxy the call setup will not work without special configuration. Figure 2 show the internal software architecture of approach A.

3.2 B: Portal with Shared Secret

There are a number of advantages by using a portal for the inclusion of phones into the e-meeting, but it also makes the architecture more complex. The same elements as mentioned in section 3.1 are used, but the phone proxy is now moved into the e-meeting portal and the session setup is done via secure communication as described in section 2.3. A new element is also introduced, a *portal SIP proxy*, which is a special kind of SIP proxy as shown below. We also assume that the session key is not known to the portal. The process then becomes as follows:

1. The user opens a user interface dialog and enters a phone number (directly or via a phone book).

2. The SIP agent sends an INVITE message to the portal SIP proxy.

3. The portal SIP proxy modifies the media description included in the INVITE so the portal becomes the new data end point instead of the client.

4. The portal SIP proxy forwards the modified INVITE to the phone SIP proxy.

5. The phone SIP proxy answers with an authorization required answer, which the portal SIP proxy either forwards back to the client or uses credentials stored locally.

6. In the former case the SIP agent retrieves the credentials needed (from local storage or by asking the user) and resends the INVITE message to the portal SIP proxy which then forwards the INVITE to the phone SIP proxy.

7. The phone SIP proxy then establishes a connection to the phone itself and sends an acknowledge message to the portal SIP proxy.

8. The portal SIP proxy then sends back the acknowledgment to the SIP agent together with a request for encryption credentials for the e-meeting.

9. The SIP agent encrypts the session credentials with the shared secret, that was earlier exchanged via SSL (see section 2.3), and sends that to the portal using a SIP extension message[3].

10. The portal then sets up the phone proxy that handles en-/decryption, transcoding and mixing of the audio.

11. Audio data can now be exchanged between the phone and the e-meeting.

12. At the end of the conversation the call is terminated by either the phone user (the phone SIP proxy sends a BYE message to the portal SIP proxy) or by the local client user (the SIP agent sends a BYE message to the portal SIP proxy which forwards it to the phone SIP proxy.

This method allows for media handling in the portal independent of the end client application, but it still means that the portal need to get a copy of the shared key used within the session. Figure 3 show the internal software architecture of approach B.

3.3 C: Portal without Shared Secret

The algorithm described in the previous section can be further enhanced where the session key does not have to be shared with the portal, but instead the inviting client re-encrypts the audio data.

Steps 1- 8 are the same as in the previous section, and it continues as:

9. The SIP agent sends a new SIP INVITE to the portal SIP proxy requesting a point-to-point session. The encryption key used for this separate session is the same as was earlier exchanged in the initial setup between the client and the portal.

10. The portal then sets up the phone proxy that handles en-/decryption, transcoding and mixing of the audio but handles audio data directly to and from the client only.

11. Termination is handled as before.

The advantage here is that the audio data will only be available to the portal while the phone call is active. As soon as it is terminated the portal will not receive any more audio data. Figure 4 show the internal software architecture of approach C.

[3]This extension message is currently not standardized and should be seen as application specific in this context.

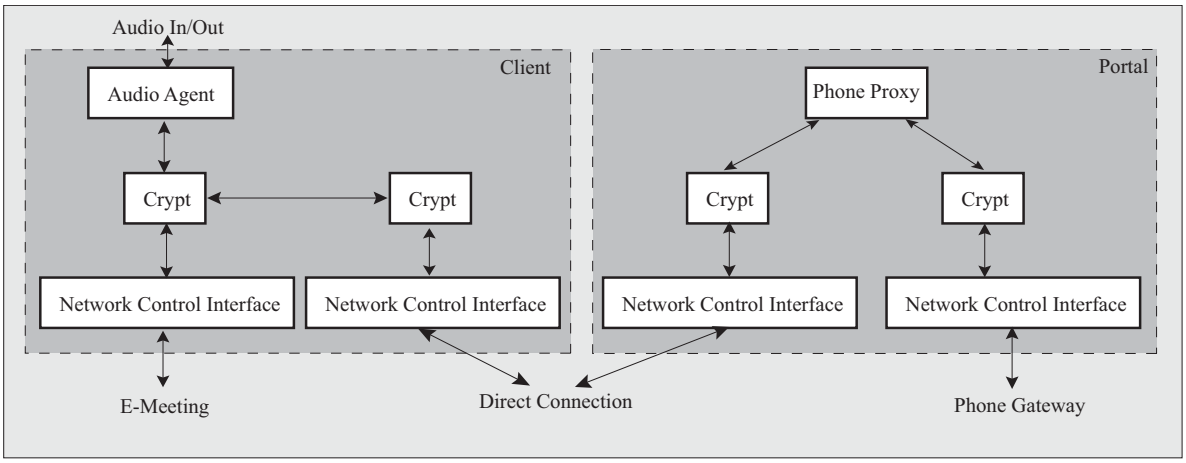

Figure 4: Internal architecture of approach C.

3.4 User Membership

A very important security related issue is how to handle the situation where the inviting user wants to leave the session. In scenario A and C this will obviously force the phone user to leave the meeting (as the traffic is passing through the user software), but in alternative B no such strict need exists. In the realization behind the architecture, it is left up to the leaving user to decide, but the default value is not to let the phone user stay behind in the meeting.

4 Performance Evaluation

The various alternatives introduce different amounts of delay and bandwidth utilization. These amounts vary depending on the audio encoding used and as reference can be assumed that PCM encoded audio is used for communication with the phone gateway. This of the most common non-proprietary audio encoding and has a data rate of 64 Kbps which translates to effective network 71 Kbps (if 40 ms frames are used). A common proprietary audio codec with good robustness and high audio quality is the GIPS iPCM-wb codec with a variable bit rate of an average 80 Kbps (89 Kbps effective). Note that all modern networked audio applications utilize silence suppression (or user controlled “click to talk”), and audio data is only sent when there actually is something to send.

4.1 Bandwidth Utilization

When a phone is included into an e-meeting, data has at least to be sent to and from the phone gateway, and as all traffic to the phone is mixed before transmission the network utilization will be 71 Kbps independent of the amount of simultaneous active audio senders.

In cases A and B no extra bandwidth is used, but in case C an extra $(n + 1) * X$ Kbps, where n is the number of active audio senders and X is the bandwidth of the audio encoding used, will be utilized between the client and the portal.

4.2 Delay

As with all synchronous communication tools delay is very important. In the architecture presented in this paper the only component that generates any significant extra delay is the audio mixer due to algorithmic delay in the audio codec.

Using a Pentium4 2.8 GHz PC an extra delay of audio sent to the phone gateway is a minimum 44 ms of which 2*20 ms is the decoding and encoding delay in the audio codec. The other 4 ms are general application handling and encryption/decryption. Note that the optional transcoding between different audio types is included in the mixing process as all audio streams, independent of type are handled in linear 16 bit encoding. On top of the 44 ms is a dynamic jitter buffer that can vary from 0 ms to 100 ms depending on packet-loss and jitter between the arrival times of the packets in the audio stream. The delay introduced by the phone gateway is not included here as it depends which type of hardware the gateway provider is using, but it is typically 20 ms is each direction as packets has to be stored and forwarded (unless the gateway is doing direct bit forward to the phone system in which case the delay can be neglected). Data from the phone gateway is only re-encoded if the default audio encoding of the session has a lower bandwidth than the one used with the phone gateway. In most cases no re-encoding will be necessary.

On top of the audio coding delay there is also a network propagation delay that obviously varies a lot between different network setups. In the evaluation presented here the network propagation delay to the phone commercial gateway from the client or the portal was measured to 22ms.

All in all, the extra delay introduced by including a phone into an e-meeting is low enough to allow users to use it for real meetings. An interesting observation is that several phone users have stated that they get better perceived audio quality when talking with users in e-meetings than if they talk with another phone user. The rationale for this is that the audio equipment on a modern computer is of higher quality than that in a normal phone.

5 Discussion

This paper presents an architecture for secure inclusion of phones into online e-meetings including three different approaches for handling coding and key management. In approach A everything is handled by the end client and the session key information does not have to leave the client. In approach B the session key is shared with a portal that does the media handling. If it is not acceptable to share the media key with the portal then approach C shows how re-encryption of the audio media is done at the client and then all audio streams are forwarded to the portal, during the duration of the phone call.

The interaction with the phone gateway is done using the Session Initiation Protocol and the paper shows how the portal can handle SIP invitations using credentials only stored at the client, while handling the actual audio data at the portal.

The reason for including a portal into the architecture is that running all the audio transcoding, mixing and decryption/encryption is dependent on what the client host is doing. I.e. if the user starts a larger program that momentarily takes a lot of CPU, then that will create glitches in the audio for the phone connected user and/or e-meeting members listening to the phone user.

In the commercial world everything is not always as perfect as it is in theory or in an academic setting. Most[4] generally available commercial SIP based phone gateways today do not support encrypted media transport, leaving a big hole in an otherwise secure online e-meeting. This leaves

[4] All that we have found actually.

the option of operating a phone gateway of your own or accepting that the mixed audio data is actually transmitted un-encrypted over the IP network between the audio proxy and the phone gateway.

An evaluation of the delay was presented and user tests have shown that the delay is low enough to make the system usable and some users perceived the resulting audio quality better than using only phones.

5.1 Implementation

All three approaches presented in this paper have been implemented into the Marratech Pro and Marratech E-Meeting Portal solution. These products are mainly developed using the Java language with about 10% of the Marratech Pro application code developed in C/C++ due to performance issues (e.g. audio and video coding).

The audio coding component is produced by Global IP Sound where the wide-band GIPS iPCM-wb codec is the most commonly used for e-meetings. Note that this codec is GIPS proprietary.

The SIP User Agent in the client and the SIP proxy in the portal are based on the Jain-SIP1.1 [13] reference implementation provided by the National Institute of Standards and Technology, NIST.

The prototype presented here has been verified and used on several different operating systems including Microsoft Windows (Windows 98 and up), Apple MacOSX, Linux and Solaris.

5.2 Future Work

In the current architecture phone users cannot call into an e-meeting because of the problems of choosing which e-meeting to dial into as well as the problem of providing credentials to enter a secure e-meeting. The approach we are working on right now is based on the user using either her phone via WAP [14] or a Java J2ME MIDP Midlet [15], or via a WWW page using a separate computer[5] for choosing which session to enter and providing credentials to the portal to be authorized to enter the meeting. The portal then sends back a pin-code (a password) and a phone number to call. The latter is the number to an incoming phone-gateway registered to forward all calls to the portal. The user calls the number and enters the pin-code provided earlier by the portal. An alternative, depending on the setup is that the portal makes a call directly to the phone via a phone gateway. This introduces the question of who will pay for the call and one further alternative is that the phone users provides phone gateway credentials to the portal (when registering initially or for each call) and that phone gateway account is used for calling the phone users. The latter might also be attractive to do depending on different call rates in different directions (i.e. it might be cheaper to call *to* the phone than *from* the phone). Storing the credentials or forwarding them to the portal might be seen as non attractive if the user does not trust the portal administrators.

Acknowledgments

The author would like to thank Hans Öhman, Marratech AB for helping with the architecture design and Per Danvind, Marratech AB with helping with implementing the audio proxy.

[5] Obviously, it would be better to use the local computer for the communication, but it might lack the right computer audio equipment and it might be used in conjunction with the phone.

This work is supported by the Centre for Distance-spanning Technology (CDT) under the VITAL project, Marratech AB, and the Swedish Research Institute for Information Technology.

References

[1] S. E. Deering, *Multicast Routing in a Datagram Internetwork*, Ph.D. thesis, Stanford University, 1991.

[2] S. McCanne and V. Jacobson, "vic: A flexible framework for packet video," in *Proceedings of ACM Multimedia*, 1995.

[3] V. Jacobson and S. McCanne, "vat - LBNL Audio Conferencing Tool," URL[6].

[4] H. Eriksson, "Mbone: The multicast backbone," *Communications of the ACM*, vol. 8, pp. 54–60, 1994.

[5] P. Parnes, K. Synnes, and D. Schefström, "mstar: Enabling collaborative applications on the internet," *Journal of IEEE Internet Computing*, September/October 2000.

[6] P. Parnes, *The mStar Environment - Scalable Distributed Teamwork using IP Multicast*, Luleå University of Technology, December 1999, Doctoral Thesis, ISSN 1402-1544 / ISRN LTU-DT–99/31–SE / NR 1999:31.

[7] "Packet-base Multimedia Communication Systems," September 1999, ITU-T Recommendation H.323.

[8] Jiann-Min Ho, Jia-Cheng Hu, and Peter Steenkiste, "A Conference Gateway Supporting Interoperability between SIP and H.323," in *Proceedings of ACM Multimedia'2001*, October 2001, pp. 421–430.

[9] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session initiation protocol," June 2002, IETF RFC3261.

[10] Baugher, McGrew, Oran, Blom, Carrara, Naslund, Norrman, "The secure real-time transport protocol," June 2002, draft-ietf-avt-srtp-05.txt, IETF work in progress.

[11] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," 1996, IETF RFC1889.

[12] M. Handley and V. Jacobson, "SDP: Session Description Protocol," April 1998, RFC2327.

[13] Sun Microsystems, Inc., "JAIN SIP API Specification," JSR-000032.

[14] Wireless Application Protocol Forum Ltd., "Wireless Application Protocol Architecture Specification," July 2001, WAP-210-WAPArch-20010712-a.

[15] Sun Microsystems, Inc., "Mobile Information Device Profile Final Specification 2.0," JSR-000118.

[6] <URL:http://www-nrg.ee.lbl.gov/vat/>