An Example of Using Presence and Availability in an Enterprise for Spontaneous, Multiparty, Multimedia Communications

Hyong Sop Shim, Chit Chung, Michael Long, Gardner Patton and Siddhartha Dalal
Applied Research, Telcordia Technologies New Jersey, USA
{hyongsop, chit, mlong, gcp, sid}@research.telcordia.com

Abstract—In today's distributed, team-oriented work environment, it is critical that team members should be able to conduct multiparty, multimedia conferencing spontaneously. Spontaneous conferencing is analogous to ad-hoc, drop-in meetings and impromptu discussions in multi-person offices. The knowledge of the presence and availability of team members can be effectively used to facilitate spontaneous conferencing. In this paper, we present SEC, a communications system designed to support spontaneous, ad-hoc conferencing in an enterprise. SEC makes an extensive and innovative use of the Session Initiation Protocol (SIP) for presence and availability management, conference control, and text messaging. We also discuss in detail the design of SEC and report on the initial usage experience with SEC services.

Index Terms—Presence and Availability, Buddy List, Voice over IP, Spontaneous Multimedia Conferences

A. INTRODUCTION

Critical to increasing the productivity of group work in today's team-oriented, distributed workplaces is the ability of group members to communicate with each other in an efficient manner. Group members frequently engage in spontaneous, multiparty communications. However, as group members become geographically distributed, ad-hoc, drop-in meetings and impromptu discussions in multi-person offices are often infeasible. One result is that group members often waste time and energy in scheduling efforts playing "phone tag."

One effective approach to address this issue is to enable team members to see the presence and availability state of each other in real time. This way, team members know when to initiate new communications and when to invite other members to ongoing communications.

Existing commercial instant messaging applications, such as AOL IM and MSN Messenger, enable a group of users to communicate based on the presence and availability state of each other. However, most of these systems are designed for public use and thus lack certain features that are critical for enterprise use. For example, in most commercial instant messaging applications, the support for voice communications is limited to one-to-one. Most systems that do support voice conferencing require advanced scheduling and thus do not effectively support the spontaneity of enterprise group communications.

In this paper, we present Spontaneous Enterprise Communications (SEC). SEC is designed to support spontaneous, ad-hoc conferencing in an enterprise. In particular, SEC enables team members to subscribe to the presence and availability state of each other and to conduct conferencing with those who are available on the fly. SEC provides both text and voice conferencing capabilities and enables conference participants to switch to using different media on demand. For a voice conference, SEC enables users to use the VoIP capability in the SEC client application, PSTN phones, or both, depending on their current communication capabilities.

SEC makes an extensive and innovative use of Session Initiation Protocol (SIP) [1] for presence and availability management, conference control, and text messaging. In this paper, we discuss in detail the design of SEC and report on the initial usage experience with SEC services.

The rest of the paper is organized as follows. Section B discusses related work. Section C defines terms and concepts critical to discussing SEC services. Section D describes functional requirements for SEC services. Section E presents SEC services from the perspective of end users. Section F discusses architectural issues in providing SEC services and presents an architectural overview. Section G describes in detail how SEC services are provided. Section H reports on the initial usage experience with SEC services. Section I discusses future work and concludes the paper.

B. RELATED WORK

Using presence and availability for group communications in distributed enterprise workplaces has been extensively studied and found to be effective [7], [8], [9], [10], [11], [12]. However, its use has mostly been for text-based communications. SEC extends the use of presence and availability in enterprise group communications by supporting both text and voice communications in an effective manner.

Since the emergence of Mbone [14] and its tools for multimedia communications [6], enabling multimedia conferencing services on the Internet has received a tremendous attention. Beginning with [5], efforts to enable multimedia communications on Mbone have generated an important body of work that is in wide use today for VoIP, e.g., Real Time Protocol (RTP) [3] and Session Description Protocol (SDP) [2]. SEC is an example of providing a particular application-level service, i.e., spontaneous conferencing, using these underlying technologies.

There are many commercial products that provide multimedia communication capabilities, e.g., CU-SeeMe [15] and Microsoft NetMeeting [24]. However, in most of these systems, the communication is limited to one-to-one. Those that do allow multiparty conferencing for both PSTN and/or VoIP users, e.g., Microsoft Exchange Conference Server [25] and WebEx [27], require conferences to be scheduled in advance (usually via a Web interface) and thus do not address the spontaneity of enterprise group communications in an efficient manner. ME.net [26] is an example of commercial services that provide spontaneous conferencing capabilities using buddy lists and Web pages. However, ME.net only supports PSTN phones and does not allow users to switch to a different communication media type on demand.

C. TERMINOLOGY

In SEC, *presence* is defined as data that indicates whether or not a user has logged into a communications system, and *availability* refers to the user's willingness to communicate with other users in the system. The presence and availability state of a user is called the *PA state* of the user. *PAL* stands for presence and availability list. A PAL is equivalent to a "buddy list." Each item on a PAL identifies a contact or "buddy" whose current PA state the owner of the PAL is interested in knowing. A PAL provides the main interface through which users access SEC conferencing services. We distinguish PALs from standard buddy lists in that a contact on a PAL may not only represent a human but also any object that is an event source in SEC. PAL is part of an ongoing work on using PA subscriptions and notifications in an distributed enterprise environment.

D. REQUIREMENTS

In this section, we describe the functional requirements on SEC. SEC is primarily designed to provide spontaneous conferencing in an enterprise. Hence, SEC should:

- **Integrate with a PAL.** Users should be able to initiate new conferences from their PALs. Users should also be able to add new participants to ongoing conferences from their PALs.
- **Integrate with enterprise directory**. An enterprise directory is the main source of contact information in an enterprise. Thus users should be able to manage their PALs using the enterprise directory, e.g., add a new contact to a PAL and/or add a new participant to an ongoing conference from a directory search result.
- **Support PSTN phones**. Today, PSTN phones are the main means of communication in an enterprise. For mobile users, cellular phones are still the only viable option for interactive voice communications. Therefore, while providing VoIP capabilities, SEC should also enable users to use their PSTN phones.
- **Allow mobility**. Today's workspace often spans multiple geographical locations, e.g., a different corporate location and "on the road." At the same time, one often needs to get in touch with his or her project members, managers, or customers from a remote location. Therefore, SEC should allow users to access its services from different locations. In addition, the communication devices available to users may change as they move from location to location. Thus SEC should enable users to use different communication devices.
- **Provide multiple types of communications media.** Different work contexts require different types of communications media. For example, for a sparse exchange of short ideas over a long period of time, text may be better suited than voice. For long, detailed discussions and presentations, voice may be a better choice. Hence, SEC should support multiple types of communications media. Currently, SEC supports text and voice. Support for other media types, e.g., images and video, and application sharing is also planned.
- **Enable dynamic change of communications media type.** The work context in which a communication takes place may dynamically change. For example, a group of users who have been asynchronously exchanging ideas in a text conference may wish to have a more interactive discussion and thus wish to switch to a voice conference adding a voice component to the text conference. Therefore, SEC should allow users to switch or add a different communications media type on demand.
- **Hide communications devices**. Users do not need to know the communication devices of those that they communicate with. Rather, users only need to specify the identities of the parties with whom they wish to communicate and the type of media, e.g., text or voice. SEC will then establish appropriate connections between the communication devices of choice.
- **Be "easy to use**." SEC should streamline the process through which end users access its services as much as possible. In particular, creating a new conference and inviting a new participant to ongoing conferences should be simple without the need to schedule and reserve communication resources.

Another important consideration is security. Different work contexts and corporate cultures may have different Security requirements. In fact, such requirements may range from "no security" to protection from all possible attacks. We are currently designing and developing a flexible security framework to be used in SEC. Issues related to providing secure communications in an enterprise environment is beyond the scope of this paper

.

E. USAGE EXAMPLE

In this section, we describe in detail SEC services from the perspective of end users. Figure 1 shows how an enterprise user creates his or her PAL from enterprise directory search results. In Figure 1, the window on the left is SEC's interface for accessing an enterprise directory, called DQ. Using this window, the user can submit a query to DQ in order to search for the contact information of other employees. Once the search results are returned and displayed in the window, the user can then select one or more entries in the window and click on the "Add to Contact List" button. This results in the selected users being added to the user's PAL. The current version of SEC does not place any restrictions on who can add whom on their PALs. However, in practice, different policies may be required, depending on corporate security policies, organizational structures, and cultures. Part of our future research efforts is to design a flexible policy framework that can be adapted to different enterprise workplaces.

In Figure 1, the window on the right is the main client interface for accessing SEC services. Using this window, the user can call a PSTN phone. In addition, the user sees the PA state changes of each contact on his or her PAL in real time. In SEC, the PA state of a contact may currently assume one of the following values: AVAILABLE, BUSY, and OFFLINE. When the PA state is “AVAILABLE,” the contact is willing to accept invitations to communicate. When the PA state is “BUSY,” the contact is not willing to accept invitations to communicate. When the PA state is “OFFLINE,” the contact is not presently logged into SEC. After logging in, the user can manually set his or her PA state via the SEC main client window.

Users can communicate only with the AVAILABLE contacts on their PALs. Using their PALs, users can instantly create new text or voice conferences or invite AVAILABLE contacts to ongoing conferences on demand.

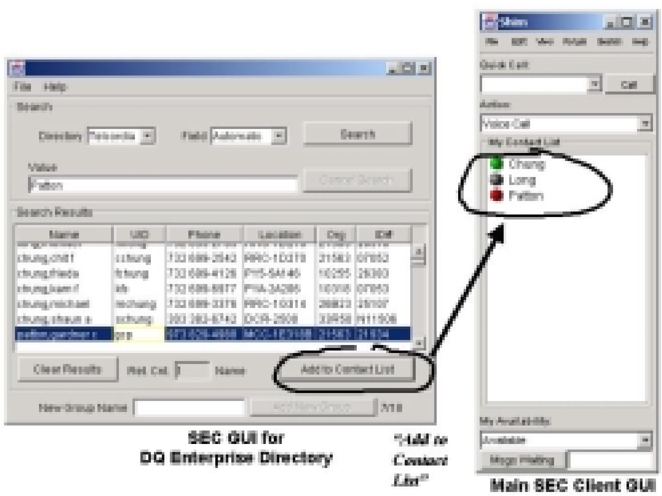


Figure 1: PAL Management with DQ Corporate Directory

Note that the PA state of a contact does not include the contact’s presence and availability on his or her specific communication device(s). This is in contrast with some current proposals for presence management [16], [19], which allow users to subscribe to and get notifications of the PA state changes on the individual communication devices of their contacts. We argue that it is easier for end users to manage their subscriptions to the PA state of other users and to control access to their own PA state at the user level than to do so at the device level. Providing a PA service at the user level still enables end users to have control over their communication devices. One approach would be to allow users to have profiles, and the system would automatically make a decision about which device(s) to use, depending on which profile is active when the user makes or receives a call. In the current implementation of SEC, each user makes a decision about which device to use on a call-by-call basis (see Section G.4).

Also note that the PA state of a contact could further be refined to include the type of communication the contact can and/or is willing to participate in, i.e., text, voice, or both. In the current implementation, SEC assumes that a user is always capable of and is willing to participate in both voice and text communications.

In Figure 2, the window on the left is the interface for a voice conference, and the window on the right is the interface for a text conference. A voice conference has a tabbed interface because users generally participate in one voice communication at a time. A text conference has its own window because users tend to participate in multiple text communications simultaneously.

In both voice and text conferences, the user can add new participants to an ongoing conference by first selecting the corresponding entries on the user’s PAL and then clicking on the “Add” button on the respective windows. Subsequently, the invited contacts are alerted and are allowed to accept or reject the user’s invitation. As part of accepting the invitation, the invited contacts decide on the communication devices to be used in the conference. Currently, for a voice conference, they can use either the VoIP capabilities of the SEC client application or a PSTN phone. Once they accept the invitation, they automatically join the conference and immediately start communicating with the other participants. In the case of a voice conference, a participant may also invite a participant by dialing a phone number from within the conference. When the called party answers the phone, s(he) is automatically placed in the conference through SEC and can immediately start communicating with the other conference participants.

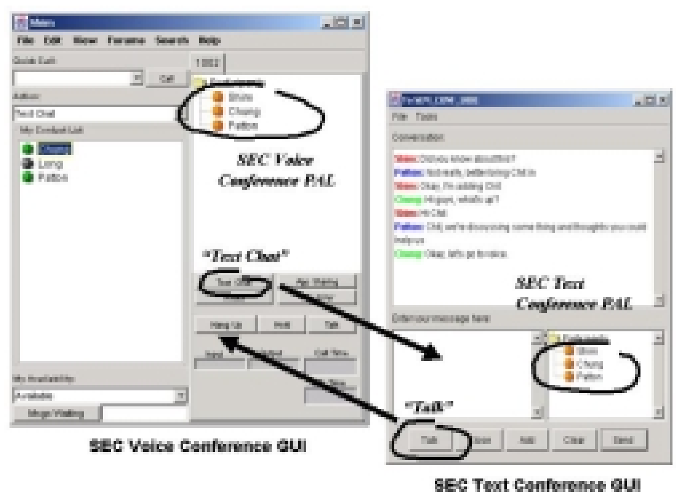
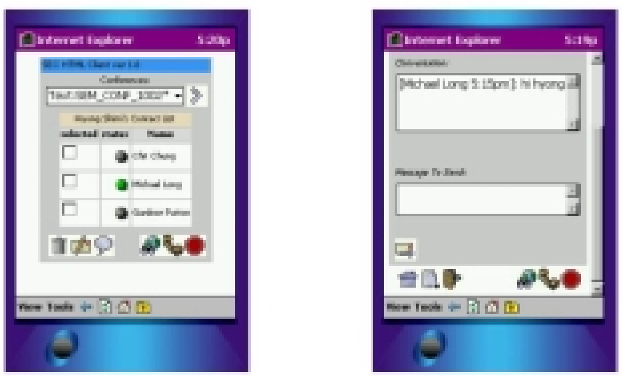


Figure 2: SEC GUI for Voice and Text Conferences

As shown in Figure 2, the windows for voice and text conferences show a participant list. This list is dynamically updated as the conference membership changes. In SEC, the conference participant list is provided as an extension to its PAL service. That is, when participants join a conference, they automatically subscribe to the PA state of the conference. Part of the PA state of the conference is its membership, and whenever the conference membership changes, SEC proactively notifies the conference participants of the new membership. This is an example of subscribing to the PA state of a non-human entity and is an effective means for conference control and management. For example, SEC could allow an authorized user to subscribe to the PA state of a conference without requiring the user to first join the conference. Then, the user could monitor the conference membership changes with minimal overhead.

Figure 2 also shows that a conference of one media type can spawn a new conference of a different media type. Hence, the participants in a text conference may dynamically switch to using voice by clicking on the "Talk" button on the text conference window, and vice versa. This ability to switch to using a new media type seamlessly, and on demand, enables conference participants to adapt to the changing context in which their communication takes place and thus increases the effectiveness of their group work.

The SEC client application shown in Figure 1 and Figure 2 is designed for desktop computers. SEC also has a Web client designed for networked Pocket PCs or PDAs running Windows CE, e.g., Compaq iPAQ. The SEC Web client enables mobile users to access and manage their PALs and participate in voice and text conferences. A mobile user who uses the SEC Web client participates in a voice conference using his or her cellular phone. Figure 3 shows the main interfaces of the SEC Web client as displayed on a Pocket PC emulator running Packet Internet Explorer for Windows CE.



SEC Client Web GUI

SEC Web GUI for Text

Figure 3: SEC Web Client

To support mobile users who do not have a networked Pocket PC or PDA, SEC could provide a voice interface to its services. For example, the "live addressbook" system [23] allows users to dial an 800 number, set their PA state, discover the PA state of their contacts, and call a contact. Through such a dial-in interface, SEC can allow those mobile users without networked PDAs or Pocket PCs to register their current phone numbers and automatically manage their PA state. For instance, when such a user receives or makes a call to a contact using SEC, SEC can notify the subscribers to the PA state of the user that the user is busy. Likewise, when the user hangs up the call, SEC can notify the subscribers that the user is now available. In this capacity, SEC would be playing the (limited) role of a client proxy for the user.

F. Architectural overview

In SEC, the concept of a *SEC client* is separated from the concept of a *communication device*. The SEC client is capable of "speaking" to SEC server components in order to access SEC services. A communication device generates and renders media, i.e., voice and text. This separation significantly increases the flexibility and availability of access to SEC services. For example, an office user who prefers the desktop phone for voice communications or whose desktop PC is not multimedia-capable, can fully utilize SEC services by running the SEC Java client application on the desktop PC and participating in voice conferences using the desktop phone. Likewise, a mobile user who has a networked PDA running Windows CE and a cell phone can run the SEC Web client on the PDA for SEC signaling and text communications and use the cell phone for voice communications.

SEC uses a centralized conference control mechanism, in which a central coordinator creates and assigns globally unique addresses, i.e., URLs, to conferences on demand. For membership control, SEC uses a SIP signaling to "dial-out" to potential conference participants. In a sense, SEC employs the model of a dial-out bridge [18] but adapts the model in order to support spontaneous conferencing. See Section G.4.

Another approach would have been to distribute the administrative tasks related to conference control among SEC clients. However, in general, managing group membership is more difficult in a distributed architecture than doing so in a centralized architecture [21]. Especially for tightly coupled conferences, in which all the participants have to agree on the full membership at all times, the group membership, in effect, can only change in lock step [13]. Therefore, a fully distributed conference control scheme may not be effective in a dynamic environment, such as a team-oriented workplace, where conferences are created in an ad-hoc manner, and the conference membership may need to change on demand and on the fly. The disadvantage of a centralized conference control scheme is that the central controller represents a single point of failure and that the scalability is limited. In a client-server architecture, server components are generally built with a high degree of fault tolerance and scalability, e.g., using multiple replicas of the same component. However, it remains a research issue to design a centralize conference control mechanism that effectively enables spontaneous conferencing in a fault-tolerant and scalable manner.

In a SEC conference, the participating clients connect to a SEC conference "bridge," and the media stream from each client is routed through this bridge to the other clients. That is, a SEC conference has a "star" configuration. Another approach is a "fully meshed" configuration, in which each client is directly connected with all the other clients in the conference, and all the media streams are directly sent from client to client. In general, without support for IP multicasting in the network, the fully-meshed configuration demands a higher network bandwidth than the star configuration. For a conference with n number of participants, the fully meshed configuration requires $n(n-1)$ network connections, whereas the star configuration requires $2n$ network connections. See Figure 4.

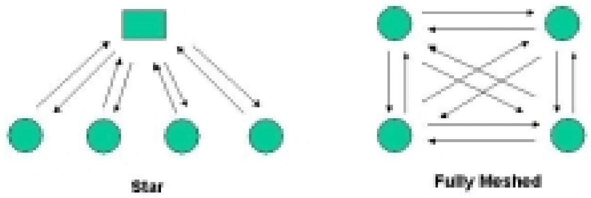


Figure 4: Example Conference Configurations

However, as in a centralized conference controller, the central bridge in the star configuration represents a single point of failure. Furthermore, the star configuration introduces a longer delay in the end-to-end delivery of voice packets than the fully meshed configuration. In SEC, because a client is connected to a conference bridge when it joins a conference, different bridges may be dynamically allocated to handle different conferences. In our initial experience with the prototype implementation of SEC, we have not experienced significant delay in conferences on our corporate intranet that spans two corporate sites, Morristown and Piscataway in New Jersey. However, a study is needed to evaluate the performance of SEC in a more general environment.

In SEC, a voice conference bridge does not mix voice streams "in the middle" for two reasons. First, because the mixing is a CPU intensive operation, it limits the scalability of the bridge. Second, it limits control over simultaneous voice streams that end points may require to provide application-specific services. For example, different users may wish to focus on different speakers. Providing this kind of service is infeasible with a mixing bridge.

Therefore, in SEC, the voice conference bridge simply routes voice packets to their destinations and leaves the task of mixing to SEC clients. For a PSTN phone, the bridge routes voice packets to a PSTN proxy that performs mixing and then sends the mixed stream to the phone. This approach takes advantage of the increasing availability of multimedia-enabled PCs with high processing power. Distributing the mixing task to clients helps the voice bridge support a larger number of simultaneous conferences than otherwise possible. This approach also allows the voice bridge to control network resource usage by limiting the number of voice packet streams for a given voice conference that it routes out to the clients.

Figure 5 shows the architectural overview of SEC. As previously described, SEC is a distributed client-server system. The server components include Communications Controller (CC), PAL Manager/Registrar, Multipoint Control Unit (MCU), Multipoint Text Control Unit (MTCU), PSTN Gateway Proxy, and HTTP Proxy. The CC, PAL Manager/Registrar, and HTTP Proxy form the main interface through which SEC clients access SEC services. The MCU, MTCU, PSTN Gateway Proxy, and HTTP Proxy are responsible for routing media streams between communicating clients.

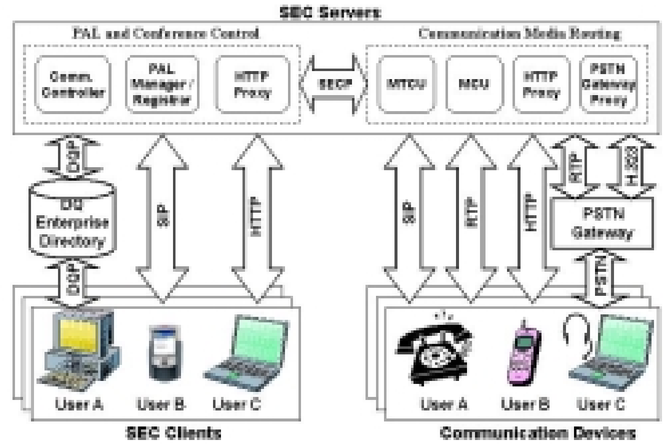


Figure 5: SEC Architectural Overview

The CC is responsible for setting up conferences and maintaining the current contact addresses of registered users. The CC also interfaces with the integrated enterprise directory.

As its name implies, the PAL Manager maintains PALs in the system. It keeps track of who is subscribing to whose PA data and who is participating in which conferences. The PAL Manager is also a system registrar; that is, a SEC client logs into the SEC system by registering with the PAL Manager. Upon successful registration, a SEC client learns from the PAL Manager the contact address of the CC to be used for its session.

The MTCU is responsible for routing text messages to appropriate SEC clients. It maintains the participant membership of each ongoing text conference in the system. A SEC client learns the contact address of the MTCU to be used for its conference(s) at conference setup time (see Section G.4).

The MCU is the SEC voice conference bridge and as previously discussed, is mainly responsible for routing voice packet streams to appropriate SEC clients. Like the MTCU, it maintains the participant membership of each ongoing voice conference in the system. A SEC client learns the contact address of the MTCU to be used for its conference(s) at conference setup time (see Section G.4).

The PSTN Gateway Proxy facilitates the participation of users using PSTN phones in multiparty SEC voice conferences. Without the PSTN Gateway Proxy, a user using a PSTN phone would be limited to 2-party voice calls. Given any PSTN phone used in a SEC voice conference, the PSTN Gateway Proxy receives the voice packet streams of the other clients in the conference from the MCU handling the conference, mixes the received streams, and sends the mixed stream to the PSTN gateway that is connected to the phone. In the reverse direction, the PSTN Gateway Proxy receives the voice packet stream from the phone via the same PSTN gateway and sends the stream to the same MCU, which in turn sends it to the other clients in the conference.

The HTTP Proxy allows users to access the SEC services using Web browsers. The HTTP Proxy receives user commands in the form of HTTP requests and transforms them into SEC operations before sending them to SEC servers.

Likewise, the HTTP Proxy receives the results of these operations from SEC servers, transforms them into HTTP responses, and then sends these responses to the user. The HTTP Proxy is designed to support mobile users who have a networked PDA running Pocket Internet Explorer. The HTTP Proxy enables such a user to enter a (cell) phone number where he or she can be reached. Subsequently, the user can initiate and participate in a SEC voice conference using the phone. The user can also initiate and participate in a SEC text conference via the HTTP Proxy.

The SEC servers and clients communicate with each other using a variety of Internet protocols for PAL management, conference control, and media transport. Specifically, SEC makes extensive use of the Session Initiation Protocol (SIP) [1] and its proposed extensions [16], [17] to register users, manage PALs, set up conferences, and transport text messages. SIP is chosen mainly because of its simplicity and flexibility. The basic SIP consists of a small number of methods and allows for incorporating application-specific semantics into its methods. In addition, SIP messages are encoded in plain text. This greatly helps streamline the testing and debugging process during implementation.

Note that the SEC server components are NOT SIP proxies. In fact, from SIP's perspective, they can be viewed as SIP end points. Also note that the PAL Manager works as a registrar in SEC only. It is not currently meant to function as a general-purpose SIP registrar.

The Real-time Transport Protocol (RTP) is used to transport voice packets between the communication devices used in voice conferences. H.323 [4] is currently used to communicate to PSTN gateways to support PSTN phones or IP phones not capable of mixing voice streams.

G. DESIGN

In this section, we discuss in detail the integration with a corporate directory, PAL management, and conference control in SEC.

1) Integration with Corporate Directory

One of the main goals in SEC is to enable enterprise employees to use their existing employee identifiers for accessing SEC services. Hence, SEC currently interfaces with a directory system, called DQ. DQ is a directory lookup service that has been developed and used in-house. The SEC client enables the user to look up other users via DQ and add search results as contacts to his or her PAL (see Figure 1). The SEC client retrieves the employee identifiers from DQ search results when adding new contacts (see Section G.3). The SEC client also allows the user to place a phone call from DQ search results.

2) Registration

When the user logs in, the user's SEC client sends a SIP REGISTER message to the PAL Manager/Registrar. In SEC, this REGISTER message contains the user's contact addresses for conference invitations (SIP INVITE), new subscription notifications (SIP SUBSCRIBE [16]), notifications of the PA state of the user's contacts (SIP NOTIFY [16]), and incoming text messages (SIP MESSAGE [17]). Note that after with registration, the user always receives SIP INVITE messages whenever the user is being invited to voice conferences. This is the case even when the user specifies use of a PSTN phone, in which case the user dynamically redirects the invitation to the phone. While this design incurs extra messaging overhead in conference control, it allows users to have a fine-grained control over how many voice conferences they are willing to participate in simultaneously and the devices to be used.

Another benefit of this approach is that it allows the user to see the full membership of the conference to which the user is being invited, even when the user uses a PSTN phone in conjunction with a PC or PDA. The conference membership data is mostly unavailable on the standard Caller-Id service today.

3) PAL Management

As discussed in Section E, SEC provides PAL management at the level of individual users, rather than individual devices. To illustrate, say User A wishes to subscribe to User B. First, User A looks up User B in DQ and then instructs the SEC client to add User B to his or her PAL. At this point, the SEC client retrieves the employee identifier of User B from the DQ search results and then composes a URL, e.g., UserB@research.telcordia.com. Subsequently, the SEC client creates a SIP SUBSCRIBE message with User B's URL in the To header field and sends the message to the PAL Manager. If User B is currently not logged in, the PAL Manager sends a 200 Ok response to the SEC client of User A. The response indicates that User B is OFFLINE.

If User B is in the system, the PAL Manager creates a new SUBSCRIBE message and sends it to the SEC client of User B. This SUBSCRIBE message is to alert User B that User B has a PA watcher, in this example, User A. However, this SUBSCRIBE message specifies the PAL Manager as the PA watcher and has the address of the PAL Manager as the NOTIFY contact. If User B has other watchers, and if the PAL Manager has already sent and received a successful response to a similar SUBSCRIBE message, the PAL Manager does not send another SUBSCRIBE message. Figure 6 graphically illustrates this design.

The goal of this design is two-fold. First, it enables the PAL Manager to keep track of all the PA subscriptions in SEC and their current status. In turn, as is discussed shortly, this removes from users the administrative burden of maintaining and downloading their PALs to new locations. Second, it facilitates an efficient implementation of the open PA subscription policy that allows every user to subscribe to all other users in SEC. By not alerting a user of all of his or her subscribers, the message traffic between the PAL Manager and the SEC client of the user is greatly reduced. If the user wishes to find out the list of his or her subscribers, the user can contact the PAL Manager.

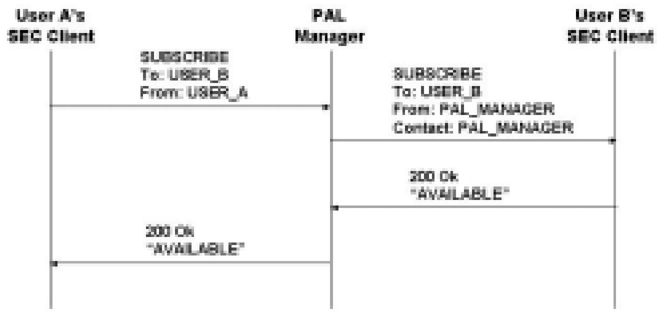


Figure 6: PA Subscription in SEC

When User B's SEC client returns a successful response to its SUBSCRIBE message, the PAL Manager records User B's current PA state, which is included in the response and sends a successful response with the same PA state to User A's SEC client. Subsequently, whenever User B's PA state changes, User B's SEC client sends a SIP NOTIFY message to the PAL Manager. The PAL Manager records User B's PA state and sends a NOTIFY message to User A's SEC client. This NOTIFY message has User B's URL in its From header.

SEC actively manages PALs on behalf ofend users in order to better support mobility. To illustrate, say User A in the previous example logs out of SEC, moves to a new terminal, and logs back into SEC from the new location. In order to download his or her PAL without any support from SEC, User A somehow has to remember the contents of the PAL and re-subscribe to the PA state of his or her contacts. A better approach would be to have User A store the PAL somewhere in the network and then download the PAL file to the new terminal. This would work well, except that User A has to remember the location of the PAL file. To improve upon this approach, the PAL Manager saves the PAL to a permanent store periodically and when User A logs out. When User A logs backs in, the PAL Manager retrieves the PAL file for the user and sends the PAL contents to the user in its response to the REGISTER message. This approach frees users from having to remember the locations of their PAL files, no matter where they are logging in.

4) Conference Set up and Management

As described in Section F, SEC employs voice/text bridges, namely MCU and MTCU, to provide its spontaneous conferencing services. In SEC, spontaneous conferencing means that the process of setting up and managing conferences should be streamlined to the extent possible. In particular, users should not have to schedule conferences or determine conference membership in advance. To this end, SEC makes an extensive use of the SIP redirection feature. In this section, we discuss in detail the conference setup and management schemes in SEC.

Specifically, SEC combines the model of users calling a conference bridge with that of a conference bridge calling users. In addition, SEC makes use of the proposed SIP SUBSCRIBE and NOTIFY methods to convey conference participant lists. That is, in SEC, each conference participant SUBSCRIBEs to the PA state of his or her conference. Henceforth, SEC NOTIFies the participant whenever the PA state of the conference changes, i.e., a new participant has joined the conference. Figure 7, Figure 8, Figure 9, and Figure 10 graphically illustrate the SEC conference set up and management schemes in the context of User A inviting User B to a conference. In these figures, the communication between the SEC server components is described in terms of the SEC proprietary protocol.

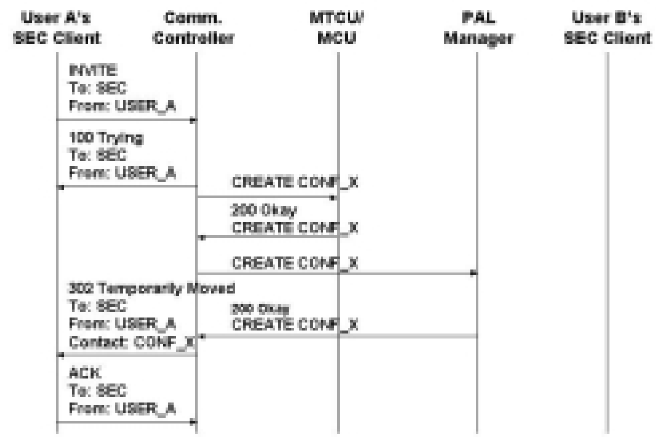


Figure 7: Creating a Conference in SEC

In SEC, when a user wishes to communicate with another user, the SEC client first creates a conference Figure 7 shows how a conference is created in SEC. In the figure, User A's SEC client sends a SIP INVITE message to the Communication Controller (CC) server. This INVITE message is addressed to a user who is a default super user, SEC. In SEC, all the INVITE messages that initiate a conference are addressed to this user. This INVITE message also includes a Require header that specifies the type of conference to be created, i.e., voice or text. Upon receiving this INVITE message, the CC creates a new conference and assigns a unique identifier, e.g., CONF_X in Figure 7). In addition, the conference is assigned an URL that is created by combining its identifier with the name of the domain where SEC is used, e.g., CONF_X@research.telcordia.com. SEC makes extensive use of conference URLs in providing its conference services.

Subsequently, the CC chooses an MTCU or MCU server to be used for this conference, and notifies the server of the new conference. The CC also notifies the PAL Manager of the new conference so that the participants of this conference can SUBSCRIBE to the PA state of the conference. Finally, the CC sends a "302 Moved Temporarily" response to the SEC client of User's A. The 302 response is a standard SIP response for redirecting calls. In SEC, the 302 response includes the URI of the new conference in its Contact: header. The SEC client of User A acknowledges the receipt of this response by sending a SIP ACK message to the CC.

Once a conference is created, the caller joins the conference. This is equivalent to a conference participant calling a PSTN bridge in order to join a conference. Figure 8 shows User A joining CONF_X. As shown in the figure, User A's SEC client sends a new SIP INVITE message to the CC. This INVITE message is addressed to the URI of CONF_X. It also contains the session description that User A wishes to use in this conference. For a voice conference, this session description either contains the IP address and port number at which User A's SEC client listens for incoming RTP/RTCP packets or the telephone number of the PSTN phone that User A wishes to use for this conference. For a text conference, it has User A's contact address to which SIP MESSAGEs should be sent.

Upon receiving the INVITE message from User A's SEC client, the CC alerts the MTCU/MCU for this conference that User A is joining the conference. In addition, the CC sends User A's session description to the MTCU/MCU. In turn, if User A wishes to use a phone, the MCU establishes an H.323 session with a PSTN Gateway Proxy, which in turn calls User A's phone via a PSTN gateway.

Meanwhile, the MTCU/MCU acknowledges to the CC that User A has joined CONF_X. This acknowledgement includes the IP address and port number to which User A's SEC client should send its voice (if User A is not using a PSTN phone) or text messages. Subsequently, the CC alerts the PAL Manager that User A has joined CONF_X. Finally, the CC sends a SIP 200 Okay response to User A's SEC client. This response contains the IP address and port number that the CC has received from the MTCU/MCU for CONF_X.

Next, User A's SEC client sends a SIP SUBSCRIBE message to the PAL Manager to subscribe to the PA state of the CONF_X conference. This SUBSCRIBE message is addressed to the URL of CONF_X. Upon receiving this SUBSCRIBE message, the PAL Manager verifies that User A is a participant of the CONF_X conference and sends a SIP 200 Okay response. This response contains the current participant list of CONF_X, i.e., User A at this point. The PAL Manager also records that it should send a SIP NOTIFY message whenever the PA state of CONF_X changes.

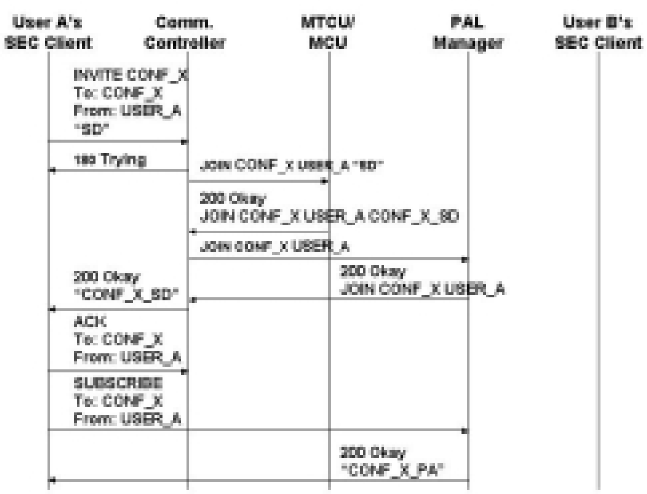


Figure 8: Caller Joining Conference in SEC

Subsequently, the SEC client of User A invites User B to CONF_X. To this end, one standard approach is that User A first calls User B and then transfers User B to CONF_X. However, this approach requires the SEC client of User A to be involved with two simultaneous calls unnecessarily and incurs extra message overhead. Therefore, SEC takes more a direct approach, in which a conference invites a potential participant on behalf of a caller. Figure 9a and Figure 9b graphically illustrate this approach.

In Figure 9a, the SEC client of User A sends a INVITE message to CONF_X at the CC. The request URI of this INVITE message contains the URL of CONF_X, but the To header contains the URL of User B. The CC interprets this message as User A inviting User B to CONF_X. Thus the CC sends an INVITE message to the SEC client of User B. This INVITE message has as its From and Contact headers the URL of CONF_X. Furthermore, this INVITE includes as part of its message body the current participant list of CONF_X, i.e., User A at this point. A better approach would be to have a header to convey this information. Such a header is not yet defined.

Upon receiving the INVITE message from the CC, the SEC client of User B sends a 200 Ok response that includes the session description of User B. The semantics of this session description is the same as that of User A. Upon receiving this response from User B, the CC alerts the appropriate MTCU/MCU that User B has joined CONF_X. This alert includes the session description of User B. Then the MTCU/MCU processes User B's session description, including, if CONF_X is a voice conference, calling the PSTN phone of User B's choice as specified in User B's session description. Subsequently, the MCU/MTCU acknowledges User B's joining CONF_X to the CC. This acknowledgement includes the IP address and port number to which User B's SEC client should send its voice (if User B is not using a PSTN phone) or text messages. Then the CC alerts the PAL Manager that User B has joined CONF_X.

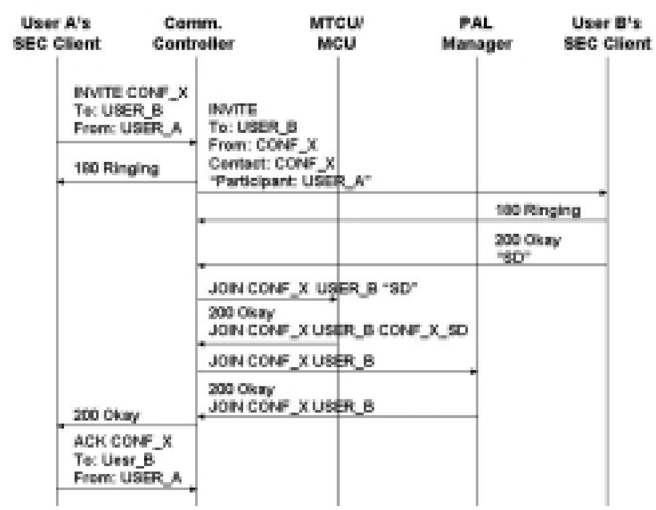


Figure 9a: Inviting User To Conference in SEC

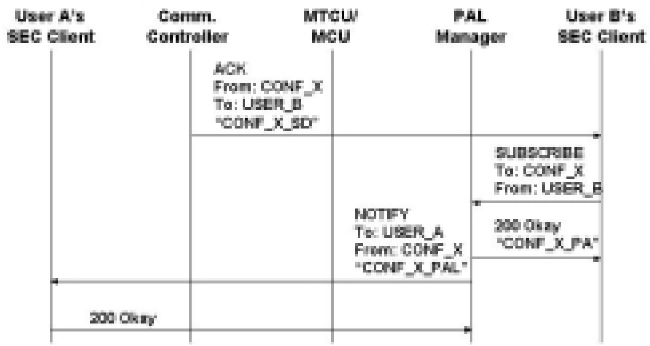


Figure 9b: Inviting User to Conference in SEC

Next, in Figure 9b, the CC sends an ACK message to User B's SEC client. If necessary, this ACK messages contains the CONF_X session description that the MTCU/MCU has sent for User B. Subsequently, User B's SEC client subscribes to the PA state of CONF_X by sending a SUBSCRIBE message to the PAL Manager. The PAL Manager sends a 200 Ok response that includes the current PA state of the CONF_X, namely User A and User B. In addition, the PAL Manager sends a NOTIFY message to User A's SEC client. This message includes the current PA state of CONF_X, namely User A and User B.

5) Conference Spawning

Spawning a conference means creating a new conference out of an existing conference. The new conference "inherits" the membership of the existing conference. In SEC, conference spawning is provided as a means of changing the type of communication media on demand, e.g., from a text conference to a voice conference and vice versa.

One issue to consider in providing conference spawning is concurrency control. Without any provision, it is possible that multiple participants will try to spawn the same conference. One approach to addressing this issue is to designate a participant as a conference administrator and enforce a policy that allows only the administrator to spawn the conference. However, in SEC, it is difficult to determine who should be the administrator, especially considering that the main idea behind spontaneous conferencing is to allow participants to freely interact with each other. Yet another approach is to have the conference spawning operation require a lock. Whoever acquires the lock first gets to spawn the conference. However, the extra step of having to acquire a lock could disrupt the fluidity of participant interactions in a spontaneous conference. In addition, always requiring a lock demands, unnecessarily, a high processing overhead, especially considering the presence of a social protocol that tends to prevent conflicting, concurrent operations in group work [20].

Therefore, SEC allows multiple participants to make a request to spawn the same conference. In order to prevent the same conference from being spawned multiple times, each request is made as originated, not from individual participants, but from the conference itself. This allows SEC to detect multiple requests to spawn the same conference as duplicates and process them accordingly.

As discussed in Section G.4, to create a conference, an INVITE is sent to the CC. The To header in this INVITE has the URL of a super user, called SEC. When spawning a conference, the From header in this INVITE contains the URL of the spawned conference. The CC honors only one such INVITE, e.g., the first one to be received, and ignores the rest. The rest of the conference spawning process is similar to the process of creating a new conference. The SEC client whose request to spawn the conference is honored is responsible for inviting the rest of the participants in the spawned conference.

6) Text Communications

SEC provides instant messaging service using the proposed SIP MESSAGE method [17]. In SEC, an instant messaging is primarily between two users. For a text communication between more than two users, a text conference is created. The process of creating a text conference is similar to that of creating a voice conference and is described in Section G.4.

In order to send a text message to the participants in a text conference, the SEC client of the sender sends a MESSAGE addressed to the URL of the conference. Upon receiving this MESSAGE, the MTCU for the conference sends to each conference participant a new MESSAGE containing the sender's message in its message body. However, one issue is how to convey the identity of the sender in the new MESSAGE. One approach is to have the sender's URL in the From header and the recipient's URL in the To header. However, this approach loses the context information that specifies that this message is sent within a text conference. Another approach is to include the URL of the conference in the From header in the new MESSAGE and the URL of the recipient in the To header. However, this approach loses the identity of the sender. Currently, SEC adapts the latter approach, including the sender URL in the message body.

H. IMPLEMENTATION AND INITIAL USAGE EXPERIENCE

The current implementation uses a SIP stack being built in-house. The CC, PAL Manager, MTCU, and SEC client are written in Java. The MCU, PSTN Gateway Proxy and the audio module embedded in the SEC client are written in C++. The audio mixing in the PSTN Gateway Proxy is completely done in software and requires no specialized hardware. The audio module uses the Microsoft DirectX technology to capture mic input and mix and play back incoming voice packets. When available, the mixing capability on sound cards is automatically utilized by DirectX. The SEC Web client is a Java Servlet that interfaces with SEC on behalf ofa user. A Cisco AS5300 gateway is used to call PSTN phones. For voice conferences, G.711 is used throughout the system Server and client machines are all PCs running either NT 4.0 Workstation/Server or Windows 2000 Professional.

SEC is in a prototype stage, and as such, no formal study has been done on the usage of SEC services. However, the current prototype has been in regular use among 4 users, who are also members of the SEC team, for the past few months. Two of the users have been actively using the AOL IM application, while the others have had a limited exposure to IM applications in general. Not surprisingly, much of use of

SEC has been related to SEC itself, e.g., reports of new bugs and bug fixes, discussions on future plans, etc.

One of the benefits of using SEC for instant messaging is security. Although the current prototype does not yet provide a security measure, the fact that instant messages do not go over the public network generates a sense of security and has been the main motivation for using SEC for exchanging work-related instant messages. Voice is preferred for long, focused discussions. Often times, in text conferences, one of the participants would inevitably send a message, saying, “go to voice?” for topics that require “a lot of explaining.” Another feature that is often used is the ability to directly call PSTN phones from the SEC client. This enables users to contact other users even when they are not logged into the system and thus greatly extends the usability of SEC.

Our usage experience also shows that the interaction with PSTN features should be improved. For example, when directly dialing a phone number in a voice conference, an answering machine or voice mail sometimes answers. In such a case, the ability to disconnect the dialed number from the conference is mandatory.

We plan to deploy SEC within our department and to conduct a formal usability, performance, and scalability study in near future. The issues of privacy and use of PA state within an enterprise organizational hierarchy should be explored.

I. FUTURE WORK AND CONCLUSIONS

In this paper, we presented SEC, a communications system designed to provide spontaneous conferencing in distributed enterprise workplaces. Spontaneous conferencing is analogous to ad-hoc, drop-in meetings and impromptu discussions in multi-person offices. SEC enables users to initiate and invite participants to ongoing conferences on demand from their PALs, where they can observe the changes to the presence and availability state of their colleagues in real time. To better support enterprise communications, SEC provides both text and voice conferencing capabilities and the ability to spontaneously switch to a different communication media type, i.e., from voice to text and from text to voice. For voice conferences, SEC supports both VoIP and PSTN phone users. SEC also provides a Web interface so that mobile users with networked PDAs and cellular phones can access SEC services. SEC makes extensive and innovative use of SIP to provide its services.

SEC is a work in progress. In particular, we plan to conduct a formal performance and scalability study of SEC’s server-based approach to multiparty communications. Given that the PSTN phone is still the device of choice for voice communication in a typical enterprise workplace, increasing the performance and scalability of the SEC PSTN Gateway Proxy is especially critical.

We also plan to include support for automatic availability management. A popular approach is to use the frequency of user actions at the keyboard and/or on the mouse. We are currently looking to improve the performance of this approach by taking into account other parameters that better represent users’ availability, such as the state of a user’s computer desktop. Another issue is access control to presence and availability in an enterprise environment. One possible approach is to hierarchically organize access rights [22] to reflect an organizational hierarchy. However, this approach may not work well for team-oriented environments, where the organization of a team may not reflect the organizational hierarch of an enterprise. We are also investigating ways to apply SEC PAL and conferencing services to better address data and document sharing needs in a distributed enterprise workplace.

REFERENCES

[1] M. Handley, H. Schulzrinn, E. Schooler and J. Rosenberg, “SIP: session initiation protocol,” Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, Mar. 1999.

[2] M. Handley and V. Jacobson, “SDP: session description protocol,” Request for Comments (Proposed Standard) 2327, Internet Engineering Task Force, Apr. 1998.

[3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: a transport protocol for real-time applications,” Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, Jan. 1996.

[4] International Telecommunication Union, “Packet based multimedia communication systems,” Recommendation H.323 Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.

[5] S. Casner and S. Deering, “First IETF Internet Audiocast,” *ACM SIGCOMM Computer Communication Review*, pp. 92-97, ACM, July 1992.

[6] V. Jacobson and S. MacCanne, “vat manual pages, LBL, UC Berkeley, CA.”

[7] E.F. Churchill. and S. Bly, Virtual Environments at Work: ongoing use of MUDs in the Workplace. *Proceedings of the 1999 International Joint Conference on Work Activities Coordination and Collaboration (WACC ’99)*, San Francisco, CA, USA. ACM Press, 1999, pp. 99-108.

[8] E.F. Churchill. and S. Bly, It’s all in the words: Supporting work activities with lightweight tools. *Proceedings of the 1999 ACM International Conference on Supporting Group Work (GROUP ’99)*, pp. 40-49, Phoenix, Arizona, USA. ACM Press, 1999,

[9] R. Evard, Collaborative Networked Communication: MUDs as System Tools. *Proceedings of the Seventh Administration Conference (LISA VII)*, pp. 1-8, Monterey, CA, Nov. 1993.

[10] M. Roseman and S. Greenberg, TeamRooms: network places for collaboration. *Proceedings of the 1996 ACM Conference on Computer-Supported Cooperative Work (CSCW ’96)*, pp. 325-333. Cambridge, MA, USA, 1996.

[11] J.H. Lee, A. Prakash, T. Jaegert, and G. Wu, Supporting multi-user, multi-applet workspaces in CBE. *Proceedings of the 1996 ACM Conference on Computer-Supported Cooperative Work (CSCW ’96)*, pp. 344-353. Cambridge, MA, USA, 1996.

[12] S. Subramanian, G.R. Malan, H.S. Shim, J.H. Lee, P. Knoop, T. Weymouth, F. Jahanian, and A. Prakash, Software architecture for the UARC Web-based collaboratory, *IEEE Internet Computing*, vol. 3, Issue 2, pp. 46-54, Mar.-Apr. 1999.

[13] K. Birman, A. Chipper, and P. Stephenson, Lightweight causal and atomic group multicast, *ACM Trans. On Computer Systems*, vol. 9, no. 3, pp. 272-314. Aug. 1991.

[14] H. Eriksson, Mbone: the Multicast Backbone, *Communications of the ACM*, vol. 87, no. 8, pp. 54-60, Aug. 1994.

[15] T. Dorcey, CU-SeeMe Desktop Video Conferencing Software, *Connexions*, vol. 9, no. 3, March 1995.

[16] J. Rosenberg, D. Willis, R. Sparks, B. Campbell, H. Schulzrinne, J. Lennox, B. Aboba, C. Huitema, D. Gurle, and D. Oran, “SIP Extensions for Presence,” Internet Draft, Internet Engineering Task Force, June 2000, Work in progress.

[17] J. Rosenberg, D. Willis, R. Sparks, B. Campbell, H. Schulzrinne, J. Lennox, B. Aboba, C. Huitema, D. Gurle, and D. Oran, “SIP Extensions for Instant Messaging,” Internet Draft, Internet Engineering Task Force, June 2000, Work in progress.

[18] J. Rosenberg and H. Schulzrinne, “Models for Multi Party Conferencing in SIP,” Internet Draft, Internet Engineering Task Force, Nov. 2000, Work in progress.
[19] M. Day, J. Rosenberg, and H. Sugano, “A model for presence and instant messaging,” Request for Comments 2778, Internet Engineering Task Force, Feb. 2000.
[20] S. Greenberg and D. Marwood, Real-time groupware as a distributed system: concurrency control and its effect on the interface, *Proceedings of the 1994 ACM Conference on Computer-Supported Cooperative Work (CSCW ’94)*, pp. 207-217, Chapel Hill, NC, 1994.
[21] J. Lauwers, T. Joseph, K. Lantz, and A. Romanow, Replicated architectures for shared window systems: a critique, *Proceedings of ACM Conference on Office Information Systems*, pp. 249-260, March 1990.
[22] H. Shen and P. Dewan, Access Control for Collaborative Environments, *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work (CSCW ’92)*, pp. 51-58, 1992.
[23] A. Milewski and T. Smith, Providing Presence Cues to Telephone Users, *Proceedings of the 2000 ACM Conference on Computer-Supported Cooperative Work (CSCW’00)*, pp. 89-96, 2000.
[24] Microsoft, http://www.microsoft.com/windows/netmeeting.
[25] Microsoft, http://www.microsoft.com/exchange/productinfo/conferencing.htm.
[26] ME.net, http://www.me.net.
[27] WebEx, http://www.webex.com.