

# Access Grid Technology Development

*Lawrence A. Rowe*

Computer Science Division – EECS  
University of California, Berkeley  
(Rowe@BMRC.Berkeley.EDU)

*James A. Basney*

National Center for Supercomputing Applications  
University of Illinois, Urbana-Champaign  
(JBasney@NCSA.UIUC.EDU)

## Summary

The Access Grid (AG) is an emerging standard for distributed collaboration. The AG collaboration model allows people at different locations to see and talk with people at other locations simultaneously and encourages sharing resources (e.g., applications, instruments, etc.). An AG node has a large tiled projection screen on which remote participants and software applications are displayed. Multiple streams of audio and video encourage seamless interaction between participants at different nodes. AG technology has been used for meetings, brainstorming sessions, distance learning (e.g., seminars, classes, training, etc.), informal gatherings after work, and sharing of remote instruments. Many organizations and individuals use the system for everyday activities and group collaboration.

Some observers propose the widespread deployment of AG technology for Scientific Collaboration (including peer review) and Crisis Management. Face-to-face meetings between peers are a critical component of the national and global scientific enterprise. Current AG software, however, is inappropriate for widespread deployment at this time. First, an AG node is complex and expensive to operate and maintain. Second, AG software does not support authentication and security. And third, the underlying communication tools (e.g., shared applications, media streaming, etc.) can be improved to support capabilities requested by current AG users.

The research and development proposed here will address these problems. A programmable automation system will be developed and tested that will eliminate the need for a local operator. The AG software will be modified to support authentication of both participants and nodes and to provide credential management services that assist participants in obtaining and using credentials for secure sessions. The streaming media tools will be modified to encrypt data using the Secure Real Time Transport Protocol being developed by the Internet Engineering Task Force (IETF).

The streaming media and application sharing tools used by the AG software will be enhanced to improve interaction and communication. An existing shared whiteboard and virtual desktop application will be integrated with the system. This software will support multiple users, multicast transport, and session archiving and playback to/from a storage system that uses the IETF Real Time Streaming Protocol. In addition, these tools will be modified to use authentication and secure communication protocols so they can be used in private meetings.

The streaming media tools will be enhanced to add features required by the automation and security work. In addition, support for omni-directional cameras and virtual cameras that can be controlled by a remote node will be added to the video tool (*vic*) and support for spatial location of sound will be added to the audio tool (*rat*). Spatial sound will be used in the automation system to provide audio cues to participants that will direct the user's attention to the current speaker. Audio cues based on spatial sound will be compared with other types of cues to direct user's attention (e.g., highlighting a window and switching source(s) displayed in a window(s)).

All software, which is developed as part of this research, will be tested at several AG nodes, integrated with the open source AG distribution maintained by the Futures Laboratory at Argonne National Laboratories, and published to the public domain.

## Statement of Work

The Access Grid (AG) is an emerging standard for distributed collaboration [AG02, Childers00]. An AG node has a large tiled projection screen on which remote participants and software applications are displayed. Multiple streams of audio and video encourage seamless interaction between participants at different nodes. AG technology has been used for meetings, brainstorming sessions, distance learning (e.g., seminars, classes, training, etc.), informal gatherings after work, and sharing of remote instruments. Over 100 AG nodes exist worldwide and more are being installed each month. Many organizations and individuals use the system for everyday activities and distributed meetings.

Some observers propose the widespread deployment of AG technology for Scientific Collaboration (including peer review) and Crisis Management. Face-to-face meetings between peers are a critical component of the national and global scientific enterprise. Traveling to one site to hold peer review meetings and to conduct day-to-day project management meetings is expensive, time-consuming, and inconvenient, particularly after 9/11. Crisis Management requires rapid collection of information and consultation with experts and decision makers at remote locations. The AG collaboration model that allows people at different locations to see and talk with people at other locations simultaneously and that encourages sharing resources is the best technology for these applications.

Current AG software, however, is inappropriate for widespread deployment at this time. First, an AG node is too complex and expensive to operate and maintain. Recent research on webcast automation and control [Machnicki02, Rui01] can be applied to reduce this complexity and cost and to improve communication in AG conferences. Second, the AG software does not support authentication and security, which is essential for confidential meetings and crisis response. Algorithms and protocols to provide authentication and security exist [Schneier96, Foster98, Baugher02], but they need to be implemented and integrated into the AG software. And third, the underlying interaction and communication tools (e.g., shared applications, media streaming, etc.) can be improved. Again, algorithms and code exists for shared whiteboard and virtual desktops, but they need to be integrated into the AG model and software.

The research and development proposed here will address these problems. The goal is to add functionality to the open source distribution of AG software produced by the Futures Lab at Argonne National Laboratory (ANL), which developed the AG model. This functionality will include the addition of: 1) automated control of AG conference resources (e.g., control and layout of windows on the projection screen, operation of a/v equipment, and control of media streams), 2) security and authentication services, and 3) improvements in the underlying streaming media and interaction technology. This development will solve two fundamental problems that limit use of AG technology for peer review and crisis and disaster response, namely, simplifying operation of the equipment and supporting private meetings. In addition, it will improve the interaction between people at different locations participating in a meeting.

Current AG collaborations require an operator at each node to control the material presented on the large projection screen and to operate a/v equipment (e.g., switching between cameras, controlling pan/tilt cameras, and controlling the audio system). The user interfaces provided to operate a node are complex and typically different at each node. AG conference participants complain that cameras do not show what the user wants to see and that the constant moving and resizing of windows is distracting. Sometimes as many as 30 windows are on the screen and it is difficult to identify who is talking or to what object the speaker is referring. We propose to develop a control system that will use standard configurations for display layouts (i.e., the number, size, and position of windows and the content displayed in them). The system will use software- and hardware-based sensors to drive rule-based heuristics to specify display layout, camera control, and operate other a/v equipment. The rules system will allow each node to customize the behavior for a particular node, virtual room, or meeting. We also will add software to introduce panoramic and omni-directional camera technology [Daniilidis02] into the AG software. This technology will allow control of remote images displayed on the local projection screen (i.e., a virtual

camera). The idea is to allow remote users to control their own virtual cameras. Designing an application model and transport format to control coding and delivery of virtual camera streams will be challenging because you want to compute views that are being displayed by a remote node while limiting the data delivered to all nodes. The idea is to restrict the information transmitted to the data needed by all current virtual cameras. We believe that local operators at each node can be eliminated by designing simple, limited function interfaces to control a node and by automating the actions of the node operator. A novice user should be able to walk into a room, identify him or herself, and join a conference either by entering commands to a web browser or speaking to the system.

The Internet Engineering Task Force (IETF) and Global Grid Forum have defined protocols and tools to implement authentication and security [Adams99, Dierks99, Linn00, Myers97, Novotny01, Tuecke02]. These protocols and tools must be incorporated into the AG software so conferences and meetings about confidential or sensitive material can be held without worrying about who is participating and whether others can observe the communication. This development will require the introduction of services to: 1) control which participants and nodes are allowed to join a session, 2) manage discovery and retrieval of public keys for participants so identities can be confirmed, and 3) manage keys for data encryption so streaming media content and application data are secure. And, the streaming media tools (i.e., audio, video and data), remote control, and data access and visualization tools must be updated too. In particular, the streaming audio and video tools must be updated to use the Secure Real Time Transport Protocol (SRTP) being standardized by the IETF and interfaced to the Grid standards for authentication and certificate management [Baugher02].

The current streaming media and interaction tools used in the AG can be substantially improved. Some of these changes involve modifications to existing software and some require the implementation of new software infrastructure and tools. The dominant communication modes used in AG collaborations are streaming audio and video, a distributed chat room (a.k.a., a MUD/MOO), distributed PowerPoint, and shared access to web pages. Most nodes install a tool like the Virtual Network Computer (*vnc*) that allows remote operation of a desktop computer [VNC02], and some research groups have experimented with RGB capture tools and shared whiteboard applications. However, the community has not yet adopted these tools and applications because the software is incomplete, immature, or inappropriate. The biggest limitations, as expressed by AG users, are the inability to observe the output and share control of applications running at a remote node and to work simultaneously on a shared drawing or document. We propose to solve these problems. Many commercial products and public domain shared whiteboard tools have been developed (e.g., Coccinella [Bengtsson02], CVW Whiteboard [CVW01], drawboard [Zielinski02], MediaBoard [Tung98], Teledraw [Rozek96], Tivoli [Pedersen93], *wb* [WB95], WBD [Highfield98], etc.). Few of these applications use multicast communication protocols and IETF standard streaming media protocols (i.e., RTP [Schulzrinne96]). And, none of them are integrated with the Grid authentication and security protocols. The specific tools required by the AG community include a shared whiteboard tool and a shared, multicast *vnc* tool. The goal is to maximize deployed functionality at minimal cost, which is why we anticipate that existing tools can be modified and integrated into the AG software. Important features these software packages must accommodate are multicast communication and recording and playback using the AG RTSP-compliant archiving system (i.e., Voyager [Stevens01]).

Lastly, the AG streaming media tools need continuing support and enhancement. The AG software uses the Mbone tools for audio streaming (i.e., *rat*) and video streaming (i.e., *vic*). Some of the enhancements required include: 1) adding virtual camera support to *vic*, 2) implementing user interface affordances to both tools (e.g., speaker identification and audio source localization), 3) integrating new CODEC's developed by others (e.g., MPEG4 video), and 4) adding support for application-level multicast protocols (e.g., source specific multicast [Holbrook99, Bhattacharyya01]). Other changes will be required to both tools to integrate the control automation system and security and authorization work proposed here and to support bridging to other conferencing systems (e.g., VRVS, H.32x, SIP, etc.). The Open Mash Consortium is the only group providing on-going support and development for these tools [OM02]. The

consortium funding ends in August 2002 so continuing support for these tools will end unless this proposal is funded.

The remainder of this proposal is organized as follows. The next section describes in detail the specific projects to be completed, the technical approach we propose to follow, and milestones for completing the work. The third section discusses project organization and management. The fourth section describes the broader impacts of the proposed activity including benefits to society, integration of research and education, enhancement of research and education, participation of underrepresented minorities, and dissemination of results. The last section describes results from prior grants.

## Technical Approach

This section describes the technical approach we will use to implement automated control, integrate authentication and security, and enhance the interaction and streaming media tools.

### ***Automated Control***

Our goal is to simplify operation of an AG node so a specially trained operator is not required. Users will still need some control over what is happening, but we believe it is possible to reduce those controls to a few simple commands that can be executed using a voice interface or pen/mouse commands on a wireless remote control device, PDA, or laptop.

Several research groups have explored the idea of automating webcast and videoconference production [AA02, Abowd99, Machnicki02, Mukhopadhyay99, Rui01, Steinmetz01]. These systems focus on controlling cameras (e.g., automatic tracking of a speaker and switching between different cameras), special-effects processors (e.g., compositing several video sources into one signal), and the media streams produced during an event. An ITU H.32x videoconferencing system multipoint control unit (MCU) supports limited types of automation. For example, an MCU conference has only one output stream, although most MCUs have video compositors that support picture-in-picture, two images side-by-side, or four tiled images. MCUs can switch between different video streams or a source composited from those streams. In most cases, a person manually operates the MCU during a conference, although nearly all MCU's support "video follows audio" source switching.

An AG conference is a different experience because participants see all remote participants all the time. In other words, an AG conference can have many more simultaneously displayed views of remote participants than in typical videoconferencing systems. Most AG nodes have three or more cameras, and all cameras are displayed at all nodes participating in a conference. The remote cameras are shown as thumbnails in a single window unless the node operator decides to display a larger image in a separate window. The operator tries to show active participants in large windows. But, keeping up with the different speakers and manually moving the cameras to show speakers in the local node is difficult. For the best results, a separate operator is used for each camera and the node operator concentrates on what is displayed on the projection screen, which for most conferences is prohibitively expensive.

We will extend previous work on videoconference and webcast automation to handle decisions made by an AG node operator during a conference. This automation will require controlling both the windows displayed on the projection screen as well as the video streams produced by the node. We developed and deployed several tools for producing and controlling webcasts that can be used to solve the AG control problem [Rowe01b]. The Director's Console (*dc*) [Yu01] allows a director to control equipment in a studio classroom (e.g., lights, projectors, cameras, audio mixers, etc.) and produce a webcast. Using *dc*, a producer/director can control the number and content of the webcast streams and the operation of the processes that produce the webcast (e.g., the archiving system, transcoders used to simulcast the program using different bandwidths, media formats, and transmission systems [Fitz01], special-effects processes [Mayer-Patel99], etc.).

Our first attempt to automate the webcast director function was the Virtual Director (*vdc*) system that automated camera switching and the structure of the webcast (e.g., webcast opening, transitions, and closing) [Machnicki02]. The *vdc* system was composed of a broadcast automation service and a question monitor service. The broadcast automation service allowed a director to specify various decisions about the webcast using an *automation specification language* (ASL). Decisions included: 1) transitions when starting and stopping a webcast, 2) switching between different video sources, and 3) adding and removing video streams to the webcast. Figure 1 shows a narrative description of an ASL script for a webcast. This script is specified as a collection of rules composed of a predicate and action code. The action code is executed when a predicate becomes true. This code can change webcast parameters such as switching video sources, adding a stream, setting a timer to trigger another event, displaying a title, and so forth. The *vdc* question monitor service detects questions from the local audience by monitoring microphones mounted in the ceiling above the audience and switching the video source for a stream to show the person asking the question. These two services are tied together by a graphical user interface (GUI) that allows manual intervention.

We compared the performance of the automation system to a professional television producer/director by having each produce a webcast for the same event. The automated system performed remarkably well.<sup>1</sup> A key concept in this system was the idea of a rules-based scripting language for specifying production decisions. This language allows development of different production styles (i.e., aesthetic decisions) and styles customized for different types of events. The *vdc* system is the only automation system being developed that allows scripting of the automation decisions. While this first attempt was promising, several problems in the design and implementation significantly limited the production styles that could be specified (e.g., events were limited to timers and spatially located audio, rule-sets could not be grouped, the behavior of the two services was not integrated, etc.). This first attempt at automation was a “quick & dirty” proof of concept. The system was successful so we want to build a more flexible system and apply it to a more challenging problem, specifically, an AG conference with multiple operators.

An Access Grid conference has a different architecture and different automation requirements than a webcast. An AG conference is a peer-to-peer webcast in which audio and video streams are sent from

1. *If the close-up view has been shown in the speaker stream for more than 45 seconds, switch to the next view where the next view is defined by the list {stage, stage, audience}.*
2. *If the stage view has been shown for more than 15 seconds, switch to the speaker close-up.*
3. *If the audience view has been shown for more than 15 seconds, switch to the speaker close-up.*
4. *Show the stage view when starting the webcast.*
5. *Show the stage view when ending the webcast for 30 seconds.*
6. *Show the presentation material being projected to the audience in the content stream.*

**Figure 1: Narrative description of a webcast automation specification**

each node to all other nodes. Figure 2 shows a large screen at the front of an AG node. The projection screen is tiled with windows, most of which are video streams displaying local or remote participants. Some windows show small images (i.e., thumbnails shown in the window on the left) of local and remote cameras, which are updated once a second, and some windows show large images of selected cameras, which are updated many times a second. The large window on the right is a local application; in this case it is a web browser. Remote participants can observe the application if a camera is aimed at the projection screen or the RGB signal is captured after being scanned converted to a video representation. Neither approach works well.

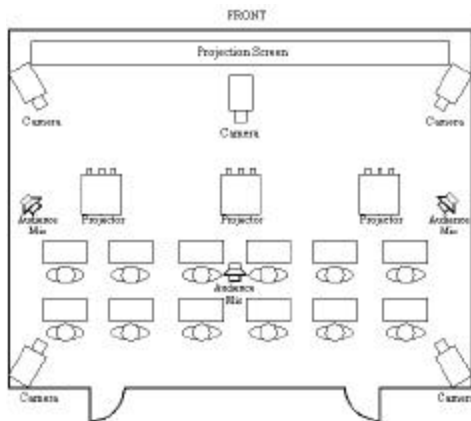
---

<sup>1</sup> The professional producer/director was impressed with the quality of the automatically produced event, and he was surprised at the simplicity of the camera switching rules. He commented, “Is that all we do?” Never the less, his production was better than the *vdc* production. The *vdc* production was still good enough for many uses.

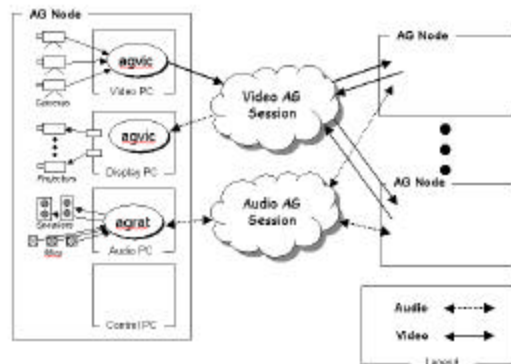
Figure 3 shows the floor plan view of a node. This node has five cameras and three microphones. Figure 4 shows the system architecture of an AG conference. Each node has several computers to handle audio, video, display, and control. An AG conference is composed of at least two IP Multicast sessions: one for video and one for audio. A node joins a conference by selecting a virtual room from a set of rooms managed by a AG venue server. Entering a virtual room causes a node to switch to the multicast sessions that define the conference for that room. Node participants see and hear participants at other nodes that are in the same virtual room.



**Figure 2: An AG node projection screen.**



**Figure 3: AG node floor plan.**



**Figure 4: AG videoconference process abstraction.**

The research we propose includes automating the layout and content of windows on the projection screen, controlling local a/v equipment and services, and exploring new models of interaction. These ideas will be discussed in the following paragraphs. One immediate need is to store configuration information about a virtual room so an operator does not have to re-organize the video and application window layout every time you enter a virtual room.

The biggest problems facing an AG node operator are: 1) deciding what large images to show, 2) where to place windows on the projection screen, and 3) positioning the pan/tilt cameras. Several people have

suggested the use of automatic layout for the display [Olson01]. Initial efforts to implement a layout algorithm have not worked well, largely because they are based on syntactic parameters (e.g., total number of video streams, number of large images being displayed, etc.) rather than incorporating semantic parameters (e.g., where a stream originates, who is talking, analysis of image and sound content, whether the conference is a lecture or project review meeting, etc.). A programmable framework is needed that will allow us to develop algorithms that can incorporate new semantic information and layout and content decisions.

Humans are well attuned to spatial cues (e.g., identifying where a sound originates or locating an object in a messy office). Spatial location and memory can be exploited by placing video windows from a particular node in the same location on the projection screen. Moreover, since most nodes produce several streams you might choose to display all of them in one area with one or more of the streams displayed as large images. The video stream displayed in one or more large windows might change depending on who is talking or which nodes had recent participation. Or, you might display one wide-angle camera view for a remote node and switch between other cameras periodically. Incorporating panorama or omnidirectional cameras will allow different nodes to display different views of the remote participants [Foote00]. Given a video stream constructed by stitching together several images, a virtual camera can be defined that displays a region in the image. Several virtual cameras can be defined on one video stream, which will allow the person viewing the stream to pan and zoom to different views.

Camera positioning is a well-studied research idea (e.g., tracking, locating the speaker, etc.) [AA02, Drucker95, Fayman98, Grove98, Kameda99, Karp93, Mukhopadhyay99]. Some vendors sell cameras with built-in tracking that work quite well [ParkerVision02], and as mentioned above most MCUs implement “video follows audio” automation. Identifying speaker location based on analyzing multiple audio input channels has also been studied [Jan95]. The research issue here is not to develop new algorithms; rather, it is to figure out what should be tracked, when tracking should be enabled, and how tracking events should be incorporated into the ASL scripting language.

Consider the following example. Suppose we have a camera that presents a wide-angle view of the audience at each site. A simple automation rule might show this view unless someone at the node is talking. The video source might be switched periodically to show other views of the audience. When someone at the node begins to talk, the view might be switched to that person using sound source localization. The person might be shown in a separate window in a different location on the screen reserved for active speakers. This window might be one of a set that contains images of the most recent speakers. Or, a region on the screen might be dedicated for speakers from a specific remote node (e.g., medium-sized images might show the remote node and a large image might show the speaker at that location). Notice that this algorithm addresses a complaint made by many AG users, specifically, that they have trouble identifying the speaker when many remote nodes are participating. Spatial repositioning of the audio output might be used to direct the attention of the local participants to a screen location where the speaker is being displayed. Other approaches to solving this focus of attention problem should be tried and compared (e.g., highlighting or enlarging the window of the speaker). We might also explore the use of technologies to identify speakers (e.g., face recognition or active badges) so titling can be automatically used to show who is talking or to influence how they are presented.

Usability testing will be required to determine when the behavior is distracting as opposed to being helpful. These algorithms must be tested by doing usability and human factors experiments to assess human responses.

The AG architecture and communication model allows more sophisticated display layouts and control patterns because many video streams can be displayed at varying sizes and quality. The mechanism we plan to use will allow different algorithms to be implemented and compared. And, different AG nodes can customize the behavior to local constraints in the room or user preferences using a different script. The mechanism we envision will have user-defined callback functions that are called when significant events

occur during the conference (e.g., a new node joins a conference, a new video stream is received, a floor-control change is signaled, etc.). These callback functions can modify the display windows directly or they might send an event to a script controlling the node.

Our implementation will build on previous work done at Berkeley and elsewhere. We will use the Open Mash streaming media toolkit to implement the automation and control systems [OM02] because it supports rapid prototyping and is very flexible.

The automation system implementation will require completion of the following tasks. The estimate of labor-months (LM) of work to complete each task is shown in square brackets.

1. [6 LM] Implement servers to control the a/v equipment in an AG node. Work on this task has already begun. We have developed many servers for controlling remote a/v equipment (e.g., AMX control computers, a/v matrix routing switchers, pan/tilt and tracking cameras, etc.) as part of our research on webcast control automation. We are currently implementing a middleware package, called **INDIVA**, that provides a high-level interface to simplify the development of applications that require remote control of a/v equipment, streaming media services (e.g., encoders, decoders, etc.), and conference resources (i.e., streams, sessions, and conferences) [Rowe02].
2. [6 LM] A synchronization toolkit is needed for the ASL scripting language. The *Nsync* toolkit developed by Bailey and Konstan has the required capability [Bailey98]. It supports temporal and non-temporal constraints, and it allows the constraints to be modified dynamically. The *Nsync* implementation uses a multi-valued logic that avoids busy waiting. *Nsync* was implemented in Tcl for a different streaming media toolkit. Konstan reports it will be relatively easy to port to the Open Mash toolkit [Konstan01]. We selected *Nsync* because it has the required facilities, an efficient open source implementation exists, and it can be rapidly integrated into the existing AG software environment.
3. [6 LM] The ASL abstractions developed for *vdc* must be re-implemented to use the *Nsync* toolkit. These abstractions include conferences, media sources, media streams, and clocks (e.g., the global conference clock and timers triggered by events).
4. [12 LM] An automation control process must be implemented for an AG node. This process will execute the ASL script for the conference. It will also receive signals from various sources (e.g., the local node joins a particular session, a new node joins the session, a media stream changes the video source, audio is received from a particular node, etc.) and execute the actions specified in the ASL script rules.
5. [6 LM] Signals between the processes at an AG node are transmitted using either the conference bus or an RPC call. The conference bus is a local IP Multicast session that the Mbone tools use to communicate useful information (e.g., time offsets for a/v synchronization, number of large windows being displayed by *vic*, etc.). New event signals must be added to existing tools (e.g., a node-joined signal must be added to the AG controller, etc.), and new services added to a node (e.g., a service to observe audio participation and identify speakers).
6. [12 LM] Node automation scripts must be implemented and tested. We anticipate this task will be an on-going, continuous improvement activity. We hope to modify the scripts and test them several times a month. New services, events, and ASL abstractions will be implemented as required.

We will deploy and test this software at Berkeley, NCSA, and ANL. Assuming the tests are successful, it will be incorporated into the ANL AG software distribution.

## Total Development Effort: 48 Labor-months

### **Security: Authentication, Integrity, and Privacy**

Authentication of individuals participating in an AG session with integrity and privacy of communication channels is required to provide a level of trust for true collaboration. We propose to develop a security infrastructure for the Access Grid, based on accepted security mechanisms in the Grid community, which will add the following features.

- ?? An **authentication service** will verify the identity of all participants.
- ?? An **authorization service** will allow session managers to control who participates in a session.
- ?? A **key management service** will allow participants to use existing Grid credentials or obtain temporary credentials for session authentication.
- ?? **Session encryption** will provide communication privacy.

Grids largely make use of Public Key Infrastructure (PKI) as the standard cross Grid authentication scheme [Foster98, Butler00]. Many government agencies in the US and abroad have well-established PKI systems which makes it an obvious choice for the AG authentication requirements. The defacto PKI implementation is the Grid Security Infrastructure (GSI) developed by the Globus Project? [Foster97]. PKI is a dual key system with one very public key, bound to an entity identified in a X.509 certificate [X509]. The other key is private and accessible only to the entity identified in the X.509 certificate. The private key is used by this entity in response to a challenge that is encrypted with the public key of the entity; only the matching private key may decrypt the challenge. While PKI meets many Grid requirements, it does not address issues concerning private or secret key management. The burden is left to the user to install and manage keys and certificates (i.e., credentials) on their computer systems.

Requiring users to manage their credentials has a number of drawbacks. First, users may not be able or willing to effectively protect their private keys from compromise or loss. Second, users may access secure services, such as the Access Grid, from many devices or locations, and distributing their private keys to each device can be inconvenient and potentially insecure. Third, users may need multiple credentials to access different secure services because of differing trust policies, further increasing the key management burden. A service that securely manages user's credentials can improve both security and usability. An online credential retrieval (OCR) service provides secure and convenient access to the credentials users need for authentication [Basney02]. To make credentials available, the service either stores them in a secure repository or generates new credentials on request.

An OCR provides a credential storage mechanism that is both more secure and more convenient than a general-purpose file system. The *MyProxy Online Credential Repository*, developed at the National Center for Supercomputing Applications, is a particular instantiation of an OCR [Novotny01]. It currently allows users to upload GSI credentials for later retrieval. The Grid community has readily accepted the *MyProxy* architecture and implementation. It is being deployed in Grids around the world, including the National Computational Science Alliance, National Partnership for Advanced Computational Infrastructure (NPACI), and NASA Information Power Grid (IPG), among others.

The GSI and *MyProxy* mechanisms will be leveraged to implement the Access Grid security infrastructure. Participants will authenticate to an AG session with GSI credentials, using standard GSI protocols. The authorization service will perform access control by allowing session managers to allow or deny participants based on their authenticated GSI identities. Integration with *MyProxy* will allow users with existing GSI credentials to use those credentials for AG authentication. An online Certificate Authority will also be provided to allow users without existing GSI credentials to easily obtain credentials for a session.

Streaming audio and video tools, which currently use the Real Time Transport Protocol (RTP), will be updated to use the Secure Real Time Transport Protocol (SRTP) to provide privacy and message authentication. SRTP assumes key distribution over an external, secure control channel, which will be provided by GSI for the AG.

The following tasks must be completed:

1. [6 LM] Prepare a detailed proposal for the AG security infrastructure and solicit feedback from the AG community.
2. [13 LM] Implement the base AG authentication infrastructure. This code will include client/server authentication software for mutual authentication between participants and nodes.
3. [3 LM] Implement key exchange for SRTP. A secure, external channel must be established to exchange keys and torn-down after keys are exchanged.
4. [8 LM] Implement authorization control. This service will allow session managers to control who can join private sessions.
5. [11 LM] Develop graphical interface components. This software will provide an interface to the user authentication service with support for retrieving credentials from *MyProxy* servers. An interface will be added to the AG node to allow a user to authenticate him or herself when they enter the physical room.
6. [8 LM] Develop a GUI interface for security management. The interface will allow private sessions to be managed by the conference owner, and it will allow users to query and set security preferences and properties.
7. [8 LM] Implement an online Certificate Authority with video-based authentication. Credentials will be provided on-demand to a session manager identified by a video stream.

**Total Development Effort: 57 Labor-months**

### ***Interaction and Streaming Media Tools Enhancements***

New tools are needed to improve interaction between participants at remote nodes and to improve the audio and video streaming tools. Proposed enhancements are discussed in the following paragraphs.

#### *1. Interaction Tools Enhancements*

A typical problem in collaboration is that a participant at an AG node wants to demonstrate a program to people at other locations. During the demonstration a remote participant wants to point at the interface to ask a question, take over the keyboard/mouse and run the program, and make notes and sketches to show to other participants. Many technologies have been developed to solve this problem including virtual desktop applications (e.g., Timbuktu, *vnc* [VNC02], etc.) and shared application toolkits. Virtual desktop applications have the advantage of supporting arbitrary applications on any platform, but they cannot be displayed to multiple remote locations unless multicast communication protocols are used (e.g., Multicast *vnc* [Ziewer92]). Shared applications written to support collaboration are better, but few of the applications used by scientists and engineers today use a shared application toolkit.

AG users today typically share access to an application by running it at one node and pointing a video camera at the projection screen to capture the images.<sup>2</sup> Remote participants cannot read details on the captured images. Multicast *vnc* solves this problem. An application is run at one node and multicast *vnc* sends the display updates to other nodes in the same conference. But, only one user can control the

---

<sup>2</sup> PowerPoint is a notable exception as several Distributed PowerPoint applications have been developed.

application. Unicast *vnc* solves the control problem by using a unicast communication channel to every node. Any user can seize the floor (i.e., the keyboard and mouse), but Unicast communication will not scale. Consequently, interaction is difficult because it is cumbersome or impossible to hand control to a remote user. And none of these solutions use the proposed authentication and security services described above.

Many groups have developed shared whiteboards, but they are not used in the AG community for a variety of reasons. The shared whiteboards developed for the Mbone community (e.g., *wb*, *wbd*, MediaBoard (*mb*), etc.) are either missing required functionality (e.g., security and authentication) or need continuing support and development. The tools developed elsewhere (e.g., Coccinella, CVW, Teledraw, etc.) are not integrated with the appropriate network protocols (e.g., IP Multicast) and archiving system used by the AG.

Two tools are needed: 1) a multiple user *vnc* and 2) a shared whiteboard application. We propose to further evaluate the available tools and select and enhance tools for the AG community. We believe an existing tool can be integrated with only modest effort to solve these problems (18-24 labor-months) and the majority of that will be adding authentication and security. *Vnc* or multicast *vnc* will be modified to support multiple pointers and a simple floor control protocol. Access to the shared session will be controlled by retrieving the appropriate certificates and checking them. And, privacy will be protected by encrypting the data packets.

The shared whiteboard tool has more difficult requirements. It must provide some mechanism to handle lost packets (i.e., reliable multicast); it must provide support for participants to join at any time and receive a correct version of the whiteboard data; it must interface with the Voyager archiving system; and, it must support authentication and security. We believe the MediaBoard [Tung98] is a good candidate for further development because it already has many of these features, but we are unsure because the code does not currently work. Some other systems (e.g., Coccinella, CVW, etc.) might provide a better foundation for further development. Consequently, we propose to fix the MediaBoard bugs and compare the functionality with other shared whiteboard applications. At that point, a plan can be developed to produce a tool for the AG community.

The lack of interaction tools is a common complaint by current AG users. A portable, secure solution that integrates well with the AG environment is needed. The following tasks will be completed:

1. [12 LM] Extend *vnc* to add multicast support (or start with Multicast *vnc*), and add multiple pointers, security enhancements, and simple floor control.
2. [12 LM] Evaluate alternative shared whiteboard applications and select one for enhancement and integration with the AG.

### **Total Development Effort: 24 Labor-months**

#### *3. Streaming Media Tool Enhancements*

The Open Mash Consortium has made numerous improvements to the video-streaming tool used by the AG (*vic*). These improvements include porting the code to new platforms (e.g., Apple Macintosh), adding support for new devices (e.g., an RGB capture board), enhancing support for selected video CODEC's (e.g., adding an H.263+ CODEC, adding support for Motion JPEG restart markers and interlaced full-sized video, etc.), adding new features (e.g., user-defined automatic window layout algorithms, titling, etc.), and fixing bugs (e.g., deallocating resources correctly so a running process can switch multicast sessions, etc.). The development plan has been to support low-latency, high quality streaming media. Rather than spend programming resources to produce the highest quality images at the lowest possible bit

rate, we have invested the minimal resources required to produce good quality at higher bit rates since the AG and many collaboration experiments are using Internet2 which has bandwidth to spare.<sup>3</sup>

We believe the following enhancements should be made to the *rat* and *vic* media streaming media tools:

1. [12 LM] Virtual Camera Interface. Several people have suggested we interface panoramic and omni-directional camera systems to *vic*. FX Palo Alto Research Laboratories has donated a *FlyCam* system [Foote00] that we propose to interface to the system. Our plan is to support the system as a video device initially and then design and implement a streaming media representation and transport format that will allow remote users to create and control virtual cameras. A key research question is how to share the limited computation at the capture computer that stitches together images and controls virtual cameras. This code will be designed to support omni-directional cameras as well.
2. [6 LM] Secure Real Time Transport Protocol (SRTP). The transport protocol used in *rat* and *vic* must be modified to use the SRTP protocol. We will integrate the public domain SRTP implementation developed by Cisco [SRTP02]. Support is also needed to retrieve keys from the venue server and check them for validity. The tools currently have interfaces for media encryption, but the code has not been optimized for performance on modern processors (e.g., to exploit multimedia instruction sets). Because AG nodes want to display as many video streams as possible at the highest possible quality, encryption and decryption must be efficient.
3. [3 LM] Restructure the *vic* code. Numerous researchers have suggested that we restructure *vic* to separate the GUI interface code from the code that captures/renders images, encodes/decodes the images, packs/unpacks the data into/from RTP packets, and transmits/receives IP packets. Several incomplete versions of this separation have been attempted, but none has been complete enough to warrant switching from *vic* to that tool. We propose to do this development. At the same time, other features should be added to the software including support to allow the compression and decompression software to run in parallel on a multiple-processor system. Currently, *vic* is a single-threaded process, and only one *vic* instance can open a multicast session. Consequently, an AG node with a multi-processor display computer cannot handle more streams than a single processor computer.
4. [5 LM] Add User Interface Affordances. Nearly everyone notes that the Mbone tools are difficult to use. Several common complaints are expressed: 1) a simple combined interface is required, 2) audio levels are wrong (e.g., one remote user is too quiet while another remote user is too loud), and 3) it is difficult to identify who is talking when there are many windows showing remote participants. We should fix these problems. Building a combined audio/video interface with controls to support participation in a large AG conference without requiring too much effort by the user is needed. Audio equalization and the introduction of spatial sound will also improve usability.
5. [3 LM] Improved CODECs and Device Support. It is likely that the AG community will continue to experiment with new audio and video CODECs and devices. Some continuing effort is required on this activity, but we should document the software abstractions better so other researchers can contribute new code more easily.
6. [1 LM] Add Source Specific Multicast Support. The networking community is moving to protocols other than IP-Multicast for n-way communication. New communication models are being discussed in the Grid community, including Source Specific Multicast (SSM) and content

---

<sup>3</sup> The RTPtv software we developed provides synchronized audio and video streaming over the Internet with quality equivalent to production television using a low cost Motion JPEG board [Delco01]. The MJPEG enhancements to *vic* allow video streams produced by RTPtv to be decoded and displayed.

delivery networks (CDN). Whatever model is eventually selected, changes will be required in the application code. At this time, we believe SSM support should be added to the Open Mash Toolkit so application developers, including the AG software, can use this technology.

**Total Development Effort: 30 Labor-months**

### **Technical Approach Summary**

The total work required is 159 labor-months, which broken down for each area is:

- Automated control system: 48 labor-months
- Security infrastructure: 57 labor-months
- Interaction tools: 24 labor-months
- Streaming media tools: 30 labor-months

Adding up the available resources (e.g., 50% time academic year for graduate students and 100% time for 2 or 3 months in the summer and 100% time for staff programmers) gives 192 LM. The details above allocate 80% of this resource. The remaining time will be spent doing documentation and providing continuing support and enhancement to the software being developed.

### **Project Management and Milestones**

This project will be distributed between students and staff at U.C. Berkeley and NCSA at the University of Illinois, Urbana-Champaign. The control automation and interaction and streaming media tools development will be done at Berkeley and the security and authorization development will be done at UIUC. The two PI's will jointly manage the entire project. We will have weekly research meetings using AG nodes at the two locations. This activity will build a common team, test the technology being developed, and force the students to deal with the results of their design and implementation decisions.

The ANL group is setting up a common source code control system for the AG software. We will use this repository for all software developed as part of this project.

Yearly project milestones are:

#### **Year 1**

- Automated control system foundations (i.e., *Nsync* integration, ASL abstractions, etc.)
- Base authentication infrastructure completed
- SRTP key exchange library
- Initial release of *vnc* tool (multicast support, floor control, RTSP archiving)
- Initial release of shared whiteboard tool with plan for further development
- Initial release of streaming media tools (SSM, virtual camera device support)

#### **Year 2**

- Automated control system (i.e., automation control process, ASL, and control scripts)
- Spatial audio integration and testing to focus attention on speaker
- Authorization control support
- GUI interface to manage session and personal authentication information
- Secure transport support in interaction and streaming media tools
- Streaming media tools (CODEC improvements, *vic* process restructuring, etc.)
- Maintenance and enhancement release of tools (enhancements for automation)

### **Year 3**

- Enhancement and testing of automation scripts and services
- Tool to manage private session authorization
- Online certificate authority with video-based authentication
- Efficient transport and remote control of virtual cameras
- Maintenance and enhancement release of tools

## **Broader Impacts of Proposed Activity**

This section discusses the broader impacts of the proposed research.

### ***Benefits to Society at Large***

Distributed collaboration and streaming media on the Internet will likely have a major impact on society over the next 5-10 years. Lecture webcasting is already being used to improve local education and to implement distance-learning programs. A lecture webcast allows students to watch lectures at a different time or location. Professional conferences over the past several years have begun to use Internet webcasts to make events available to a larger audience. And, videoconferencing is now widely accepted in the business community for meetings. The telecommunications industry videoconferencing standard (i.e., H.32X) and the research community defacto standards (i.e., the Internet Mbone tools used by the Access Grid and VRVS [VRVS02]) are being widely adopted. In fact, most research collaboration systems attempt to bridge between these systems so people can participate with their preferred technology.

The biggest problems limiting the use of these technologies are access, cost, complexity, and security. Currently, an AG node is too expensive and it requires too many people to operate. Conferences that require privacy cannot use AG technology. The cost problem is changing as both computers and a/v equipment declines in price. The operational complexity and security problems will be addressed by the research proposed here.

### ***Integrate Research and Education***

Our research program integrates research and education in several ways. First, we have used Internet streaming media in classes to bring new opportunities to local students and to make some aspects of the Berkeley undergraduate and graduate experience available to a wider audience.

Prof. Rowe's research group at Berkeley developed and deployed a lecture webcasting system, called the Berkeley Internet Broadcasting System (BIBS), which allows students to view lectures live or on-demand. Fifteen classes with an enrollment of over 4,500 students were webcast in the Spring 2001 semester. Students played roughly 20,000 lectures each month. The BIBS system is typically used to replay segments of a lecture when a student is studying for exams. The BIBS lectures are published on the web with free access to anyone on the Internet. Analyzing media and web server access logs shows that students from high school through college and people from around the world have accessed the lectures. More details on the design, implementation, and use of the system are given in a report produced before it was moved to a campus service organization for continued operation [Rowe01a].

Second, we always use undergraduate research assistants on our projects. Our research groups are typically 50% undergraduate and 50% graduate students. Often, the undergraduates start out in a volunteer position doing webcast production, A/V equipment installation and operation, and simple programming tasks. An important element of our weekly research meetings is continuing education for undergraduate and graduate students on topics related to software development and management, as well as the technical topics related to the research.

Lastly, Prof. Rowe teaches classes on multimedia applications and systems. Every third year, he offers a freshman seminar that introduces students to multimedia authoring [CS39]. The emphasis is on

continuous media (e.g., streaming audio, video, and animations) so students are introduced to the data representations and tools for capturing and editing this media. He also teaches an advanced graduate class on multimedia systems and applications [CS294]. This class covers material on hardware and software for continuous media (e.g., media representations and capture, compression, and transmission of media streams), streaming media middleware, network protocols, and distributed multicast applications (e.g., video-on-demand, webcasting, and distributed collaboration). A few advanced undergraduates take the course, but the majority of students enrolled are graduate students seeking experience with the technology and problems for research projects.

### ***Enhance Research & Education Infrastructure***

We are aggressively deploying lecture webcasting and AG technology at Berkeley for research and education. The NSF Equipment Grant (CDA-9512332) described below was used to setup a multimedia-authoring laboratory, to acquire a high-speed multicast enabled network, to create a broadcast center with an A/V routing switcher and equipment, and to create several studio classrooms. The Center for Information Technology Research in the Interest of Society (CITRIS) at Berkeley funded the construction of an AG node that is being used for production and limited research experimentation.

The research proposed here will develop software to reduce the complexity and cost of participating in an AG conference. Most of the software developed at Berkeley over the past seven years is being used to produce webcast events (e.g., seminars, conferences, meetings, etc.). We fully expect to use the automation and security systems developed as part of this research in the future for distance learning and distributed, multi-disciplinary research projects. UC Davis has already installed an AG node, which is being used for CITRIS meetings. And, we recently met with people from UCSF and UC Santa Cruz to demonstrate the AG technology. They are considering using it for major interdisciplinary research projects that are also distributed geographically.

### ***Participation of Underrepresented Minorities***

Prof. Rowe has supervised a student each summer for the past several years as part of the EECS Department *Summer Undergraduate Program in Engineering Research at Berkeley (SUPERB)*. This NSF-funded program, under the direction of Dr. Sheila Humphreys, brings underrepresented minority students to campus for a research experience [Humphreys97]. Each student is given a small research project and he or she is invited to participate in all group activities. An advanced graduate student acts as the day-to-day mentor for the student. At the end of the summer, the student gives a public presentation to the EECS community about their research. SUPERB students learn about research and independent study as well as being exposed to the Berkeley environment.

### ***Dissemination of Results***

Research results will be published through technical reports, conference papers, and journal publications. We also publish source code for all applications and systems we develop. We will participate in the Global Grid Forum to contribute code and other material to the Grid standardization process. We also will participate in the annual AG Workshops.

We participate in several on-going conferences that focus on the development and dissemination of new results involving multimedia, high-performance communication, and image processing (e.g., NLANR/Internet2/Joint Techs Workshops – see <http://www.ncne.nlanr.net/training/>, ACM Multimedia Conferences – see <http://www.acm.org/sigmm>, the SPIE/IS&T Photonics West Symposium on Electronic Imaging, Science, and Technology – see <http://www.spie.org/conferences/>, etc.).

For the past seven years Prof. Rowe has webcast the Berkeley MIG Seminar at which researchers from academia and industry present talks on recent results (see <http://bmrc.berkeley.edu/courseware/mig>). Most seminars are available for on-demand replay. We are not webcasting the seminar this year because

he was on sabbatical. The seminar will begin again in Fall 2002. In addition, other classes, workshops, and seminars are being webcast.

## Prior Awards

This section discusses results from prior NSF awards.

### 1. **Acquisition of Digital Media Storage and Networking Hardware (Rowe)** (Academic Research Infrastructure Grant CDA-9512332, 9/95-6/01)

This grant supported the acquisition and installation of a high-speed computer network, a large digital media storage system, and streaming media servers. In addition, the grant provided partial support for the establishment of two multimedia-authoring laboratories and four studio classrooms. The high-speed network uses a Gigabit Ethernet backbone to connect 10/100-switched Ethernet border devices in four buildings. Classes and individual research projects use the multimedia-authoring labs to produce web material and to edit audio/video titles for distribution through videotape and Internet streaming. Internet webcasts are produced for classes, seminars, and special events using the studio classrooms. A streaming media server, web server, and tertiary storage system were acquired to store and serve content. The high-speed network connects these servers and facilities together and connects them to Internet2.

The Berkeley Multimedia Research Center (BMRC), which is a campus-wide interdisciplinary research group focused on exploiting multimedia in research and education, submitted the grant and oversaw the installation and operation of the system. Prof. Rowe is the founding Director of BMRC. Over 200 faculty and students used the equipment purchased with this grant. In addition, several tens of thousands of people from Berkeley and elsewhere played BIBS class lectures, described above. These webcasts were produced and served with equipment partially supported by this grant. This system continues to be used and enhanced. The multimedia-authoring laboratories and studio classrooms continue to be used to produce content. The storage systems and content servers are also continuing to be used to publish material. The BMRC/CITRIS AG node uses the network acquired with funding from this grant.

A primary focus of BMRC is to support the use of streaming media including audio and video in multimedia titles and in live lectures and performances. Some specific examples are "Conversations with History," "Biotechnology at 25," "ACM SIGCHI Video Demonstration Database," "An Intermediate Hindi Reader," and "Sound of 19<sup>th</sup> Century California Sheet Music." All of these titles exploit streaming audio and/or video to present information and ideas.

Numerous refereed journal and conference papers were published that were supported in part by this grant. Looking at the BMRC website over 25 papers were published between 1995-2001. In addition, many graduate students used the equipment in their MS and PhD research. Counting three faculty in EECS (i.e., Prof. Landay, Rowe, and Zakhor), 9 PhD and 29 MS degrees were completed.

### 2. **Continuous Media Middleware Toolkit (Rowe)** (Internet Technologies Program Grant ANI-9907994, 7/99-6/02)

Many researchers doing work on distributed multimedia systems and applications develop software toolkits to prototype and test their algorithms and applications. The Mbone tools were developed in the early 1990's to support multicast streaming audio and video on the Internet. Prof. Steve McCanne incorporated these tools into an extensible toolkit, called *Mash* [McCanne97]. Prof. McCanne left academia, so the toolkit was left unsupported. This grant funded continued development and support for the toolkit, now called *Open Mash*. In addition, the grant funded development of new services and applications and the production of seminar webcasts, workshops, and documentation. Six major software releases were delivered. Over 6,500 copies of the software have been downloaded from Source Forge in the past 12 months and currently Open Mash is in the top 20% of programs ordered by the number of software downloads. An REU supplement was also funded for this grant that supported two undergraduate students (summer 2000).

## References

- [Abowd99] G. Abowd, "Classroom 2000: An experiment with the instrumentation of a living educational environment," *IBM Systems Journal*, vol.38, (no.4), IBM, 1999.
- [Adams99] C. Adams and S. Ferrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols," IETF RFC 2510, March 1999.
- [AA02] AutoAuditorium, <http://www.autoauditorium.com/>, 2002.
- [AG02] Access Grid, <http://www.accessgrid.org/>, 2002.
- [Bailey98] B.P. Bailey, J.A. Konstan, R. Cooley and M. Dejong, "Nsync - A Toolkit for Building Interactive Multimedia Presentations," *Proceedings of the Sixth ACM International Conference on Multimedia*, Bristol England, November 1999, pp. 257-266.
- [Baugher02] M. Baugher, et.al., "The Secure Real Time Transport Protocol," IETF RFC, <http://www.ietf.org/internet-drafts/draft-ietf-avt-srtp-04.txt>, May 2002.
- [Basney02] J. Basney, "GSI Online Credential Retrieval - Requirements," Global Grid Forum GWD-I (Informational), February 2002.
- [Bengtsson02] Mats Bengtsson, "The Coccinella," <http://hem.fyristorg.com/matben/>, 2002.
- [Bhattacharyya01] S. Bhattacharyya, et.al., "An Overview of Source-Specific Multicast (SSM) Deployment," IETF RFC, <http://sith.maoz.com/SSM/draft-ietf-ssm-overview-02.txt>, December 2001.
- [Butler00] R. Butler, D. Engert, I. Foster, C. Kesselman, and S. Tuecke, "A National-Scale Authentication Infrastructure," *IEEE Computer*, Vol. 33, pp. 60-66, 2000.
- [Childers00] L. Childers, et.al., "Access Grid: Immersive Group-to-Group Collaborative Visualization," *Proceedings of the Fourth International Immersive Projection Technology Workshop*, June 19-20, 2000.
- [CS39] Internet Media, <http://bmrc.berkeley.edu/courseware/cs39/>, undergraduate class, EECS Department, U.C. Berkeley, December 2000.
- [CS294] "Multimedia Systems and Applications," graduate class, EECS Department, U.C. Berkeley, <http://bmrc.berkeley.edu/courseware/cs294/>, December 1999.
- [CVW01] CVW Whiteboard, <http://cvw.sourceforge.net/>, September 2001.
- [Daniilidis02] K. Daniilidis, Omnidirection Vision Page, <http://www.cis.upenn.edu/~kostas/omni.html>, 2002.
- [Delco01] M. Delco, "Production Quality Internet Television," Masters Project Report, EECS Department, U.C. Berkeley, August 2001. Also available at <http://bmrc.berkeley.edu/papers/2001-161/>.
- [Dierks99] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," IETF RFC 2246 (Standards Track), January 1999.
- [Drucker95] S. Drucker and D. Zeltzer, "Camdroid: A System for Implementing Intelligent Camera Control," *Proceedings 1995 symposium on interactive 3D graphics*, Monterey, CA, 1995.
- [Fayman98] J.A. Fayman, O. Sudarsky, and E. Rivlin. "Zoom Tracking," *Proceedings of the 1998 IEEE international conference on robotics and automation*, Leuven, Belgium, May 1998.

- [Fitz01] T. Fitz, "tgw: A Webcast Transcoding Gateway," Masters Project Report, EECS Department, U.C. Berkeley, December 2001. Also available at <http://bmrc.berkeley.edu/papers/2001-162/>.
- [Foote00] J. Foote and D. Kimber, "FlyCam: Practical Panoramic Video and Automatic Camera Control," *Proc. IEEE International Conference on Multimedia and Expo*, August 2000.
- [Foster98] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A Security Architecture for Computational Grids," *Proceedings of the 5th ACM Conference on Computer and Communications Security*, 1998.
- [Foster97] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *International Journal of Supercomputing Applications*, 1997.
- [Grove98] T.D. Grove, K.D. Baker, and T.N. Tan, "Colour Based Object Tracking," *Proceedings of the fourteenth international conference on pattern recognition*, Brisbane, Australia, August 1998.
- [Highfield98] J. Highfield, "WBD: Whiteboard," <http://www-mice.cs.ucl.ac.uk/multimedia/software/wbd/>, 1998.
- [Holbrook99] H.W. Holbrook and D.R. Cheriton, "IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications," *Proceedings of the ACM SIGCOMM'99*, Cambridge, Massachusetts, USA, 1999.
- [Humphreys97] S. Humphreys, "EECS NSF Research Experience for Undergraduates: SUPERB Paper," EECS Department, U.C. Berkeley, September 1997. Also available at <http://www.EECS.Berkeley.EDU/~humphrys/FIE.html>.
- [Jan95] E.E. Jan and J.L. Flanagan, "Microphone Arrays for Speech Processing," Invited paper of IEEE International Symposium on Signals, Systems and Electronics, October 1995.
- [Kameda99] Y. Kameda, H. Miyazaki, and M. Minoh, "A Live Video Imaging for Multiple Users," *Proceedings IEEE international conference on multimedia computing and systems (vol 2)*, Florence, Italy, 1999.
- [Karp93] P. Karp and S. Feiner, "Automated Presentation Planning of Animation Using Task Decomposition with Heuristic Reasoning," *Proceedings graphics interface 93*, Toronto, Ontario, Canada, 1993.
- [Konstan01] J. Konstan, personal communication, October 2001.
- [Linn00] J. Linn, "Generic Security Service Application Program Interface Version 2, Update 1," IETF RFC 2743 (Standards Track), January 2000.
- [Machnicki02] E. Machnicki and L.A. Rowe, "Virtual Director: Automating a Webcast," *SPIE/IS&T Multimedia Computing and Networking Conference*, Vol. 4673, San Jose CA, January 2002.
- [Mayer-Patel99] K. Mayer-Patel and L.A. Rowe, "A Multicast Control Scheme For Parallel Software-only Video Effects Processing," *Proceedings of the seventh ACM international conference on Multimedia*, Orlando, FL, November 1999.
- [McCanne97] S. McCanne, et.al., "Toward a Common Infrastructure for Multimedia-Networking Middleware," *Proceedings of the seventh international workshop on network and operating systems support for digital audio and video*, St. Louis, MO, May 1997.

- [Mukhopadhyay99] S. Mukhopadhyay and B. Smith, "Passive Capture and Structuring of Lectures," *Proceedings of the Seventh ACM International Conference on Multimedia*, Orlando, FL, October 1999.
- [Myers97] J. Myers, "Simple Authentication and Security Layer (SASL)," IETF RFC 2222 (Standards Track), October 1997.
- [Novotny01] J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy," *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*, IEEE Press, August 2001.
- [Olson01] R. Olson, personal communication, 2001.
- [OM02] Open Mash, <http://www.openmash.org/>, 2002.
- [Pedersen93] E.R. Pedersen, K. McCall, T. Moran, and F.G. Halasz, "Tivoli: An Electronic Whiteboard for Informal Work-group Meetings," *Proc. InterCHI '93*, pp. 391-398.
- [Rowe02] L.A. Rowe, W.-T. Ooi, and P. Pletcher, "INDIVA: Distributed Streaming Media and Equipment Control Middleware," submitted for publication, April 2002. Also available at <http://www.openmash.org/resources/pubs/2002/164/>.
- [Rowe01a] L.A. Rowe, et.al., "BIBS: A Lecture Webcasting System," *BMRC Technical Report*, TR 2001-160, June 2001. Also available as <http://bmrc.berkeley.edu/papers/bibs-report.html>.
- [Rowe01b] L.A. Rowe, "Streaming Media Middleware is more than Streaming Media," *International Workshop on Multimedia Middleware*, held in conjunction with the *Ninth ACM International Conference on Multimedia*, Ottawa, Canada, October 2001. Also available as <http://bmrc.berkeley.edu/papers/2001/159/>.
- [Rozek96] A. Rozek, "TeleDraw," <http://www.uni-stuttgart.de/rus/Projekte/MERCI/MERCI/TeleDraw/Info.html>, 1996.
- [Rui01] Y. Rui, et.al., "Building an Intelligent Camera Management System," *Proceedings of the ninth ACM international conference on Multimedia*, Ottawa Canada, October, 2001.
- [Schneier96] B. Schneier, "Applied Cryptography (2<sup>nd</sup> Edition)," John Wiley & Sons, Inc, 1996, ISBN 0-471-11709-9.
- [Schulzrinne96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 1889 (Standards Track), January 1996.
- [SRTP02] *libsrtplib*: A library implementing Secure RTP and the Universal Security Transform, <http://srtplib.sourceforge.net/>, May 2002.
- [Steinmetz01] A. Steinmetz and M. Kienzle, "The e-Seminar Lecture Recording and Distribution System," *SPIE/IS&T Multimedia Computing and Networking Conference*, January 2001.
- [Stevens01] R. Stevens, personal communication, May 2001.
- [Tuecke02] S. Tuecke, D. Engert, I. Foster, V. Welch, M. Thompson, L. Pearlman, and C. Kesselman, "Internet X.509 Public Key Infrastructure Proxy Certificate Profile," Draft-ietf-pkix-proxy-02.txt, February 2002.
- [Tung98] T.-L. Tung, "Mediaboard: A Shared Whiteboard Application for the Mbone." Master's thesis, University of California, Berkeley, Dec. 1997. Available at <http://www.openmash.org/resources/pubs/papers/tecklee-masters.ps>.
- [VNC02] Virtual Network Computing (VNC), <http://www.uk.research.att.com/vnc/>, 2002.
- [VRVS02] Virtual Rooms Video Conferencing System, <http://www.vrvs.org/>, 2002.

- [WB95] LBL Shared Whiteboard Tool (wb),  
<http://www-mice.cs.ucl.ac.uk/multimedia/software/wb/>, 1998.
- [X509] ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1997.
- [Yu01] T.P. Yu, et.al., “*dc*: A Live Webcast Control System,” *SPIE/IS&T Multimedia Computing and Networking Conference*, Vol. 4312, San Jose CA, January 2001.
- [Zielinski02] T. Zielinski, “drawboard,” <http://sourceforge.net/projects/drawboard/>, April 2002.
- [Ziewer92] P. Ziewer, “MulticastVNC,” <http://www.informatik.uni-trier.de/~ziewer/MulticastVNC/>, 2002.