

# Design and Development of a SIP-based Video Conferencing Application

Bengisu Tulu, Tarun Abhichandani, Samir Chatterjee\*, and Haiqing Li

Network Convergence Laboratory  
School of Information Science  
Claremont Graduate University  
Claremont, CA 91711, USA.

{bengisu.tulu, tarun.abhichandani, samir.chatterjee,  
haiqing.li}@cgu.edu

**Abstract.** Media communication using SIP is providing us with capabilities to architect applications over ubiquitous platforms including the Internet. Although there are certain attempts to make media-enabled applications, there exists a need to deploy security and directory features that constitute middleware services. The software application, described in the paper, is part of a research initiative in Internet2 community to deploy middleware services on video conferencing application. The paper describes the architecture of a Java-based SIP Client and the results of interoperability tests between four SIP user agents. Efforts are being made to enable the application with secured middleware features.

## 1 Introduction

There is a growing trend to use multimedia communications over IP-based networks including the global Internet. A few organizations have successfully deployed Voice over IP (VoIP) while others are experimenting with technology and organizational issues including justifying a business case. While VoIP has a head start, video conferencing over IP-based networks is relatively new. Several organizations intend to use video conferencing for collaboration, remote work and virtual meetings. In particular the higher education community has plans to deploy video conferencing solutions over Internet2 [1].

For such applications to work, we need signaling protocols as well as media handling capabilities. SIP [2] and H.323 [3] have been used for VoIP with SIP gaining popularity as a flexible session oriented protocol approved by the IETF. However, in the video conferencing space, we could not find many academic or commercial applications<sup>1</sup> that use SIP. Most commercial video systems use H.323 protocol over ISDN lines. Only recently have we started to see the migration of these products to IP-based

---

\* This material is based upon work supported by the National Science Foundation under Grant No. 022710. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

<sup>1</sup> MSN Messenger is a SIP client from Microsoft.

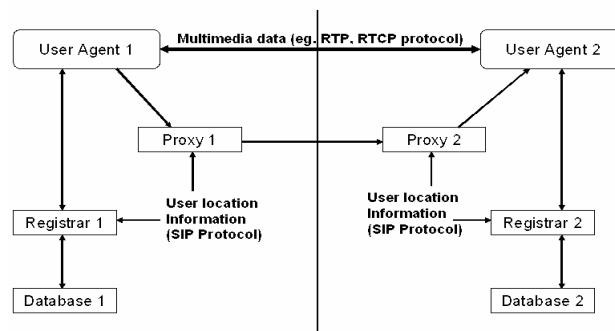
M.M. Freire, P. Lorenz, M.M.-O. Lee (Eds.): HSNMC 2003, LNCS 2720, pp. 503-512, 2003.  
© Springer-Verlag Berlin Heidelberg 2003

networks. Not only is there a need to develop and deploy SIP-based video conferencing applications, there are several requirements within the higher education community that must be met. These requirements include the ability to search directories for finding SIP users, enterprise-level authentication, and having proper authorization policies in place, which facilitates inter-campus video communications. Privacy and confidentiality of users is also needed. Moreover proper accounting and billing is an integral part to manage a converged network with voice and video applications.

In this paper, we present the design, architecture and implementation of a SIP-based video conferencing application. We also discuss our experience of deploying the client and providing service through our lab. We point out certain performance features of the client and finally conclude with a discussion of future work.

## 2 Application Design and Architecture

Session Initiation Protocol (SIP) is the Internet Engineering Task Force (IETF) standard for IP Telephony. It is an application layer control protocol that can create, modify, and terminate multimedia sessions [2]. Different types of entities are defined in SIP: user agents, proxy servers, redirect servers, and registrar servers. Figure 1 shows a simple SIP call flow including these entities.



**Fig. 1.** A typical SIP configuration [Modified from 4].

Network Convergence Lab (NCL) at Claremont Graduate University (CGU) is hosting two sip proxies: one commercial proxy from Dynamicsoft, and other open source proxy from Vovida. Dynamicsoft proxy is used as the main proxy for the test bed CGU offers to the Internet2 community. Vovida proxy is used for testing purposes. Both proxies run on the default port 5060 however, the registrar ports differ: 5070 for Vovida and 6060 for Dynamicsoft registrar.

Although there are open source SIP stacks such as Vovida and NIST, Dynamicsoft commercial stack was used to develop CGUsipClientv1.1. Dynamicsoft provides a comprehensive SIP stack including all the authentication mechanisms included in the latest RFC [2] which are not available in open source user agent SIP stacks. Dynamicsoft stack also provides multiple levels of API that allows applica-

tions with various complexity levels. The architecture of the Dynamicsoft user agent SIP stack is provided in Figure 2 labelled as Dynamicsoft Architecture.

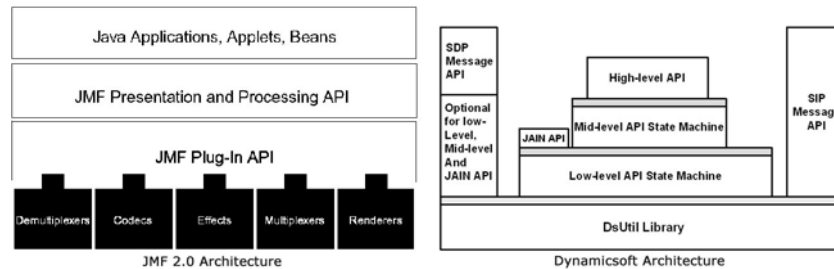


Fig. 2. JMF and Dynamicsoft Architectures

Dynamicsoft proxy and registrar provides authentication for users. Authentication type, which could be basic or digest [5], is assigned to each user during the registration process. For each registration request, registrar challenges the users for authentication based on the authentication type. The user agent is responsible for requesting the necessary information from the user and sending it to the registrar for authentication. Proxy also challenges each user for all requests except ACK and BYE message. If a user agent is challenged, it would have to resubmit its request with credentials, the request and credentials will be verified by proxy against those in the location server [6]. Further, location server would verify against the database in which it stores the details of users.

Java Media Framework (JMF) 2.1.1 Sun libraries were used to develop a SIP based video conferencing application. JMF 1.0 API (the Java Media Player API) enabled programmers to develop Java programs that presented time-based media. JMF 2.0 API extended the framework to provide support for capturing and storing media data, controlling the type of processing that is performed during playback, and performing custom processing on media data streams [7]. JMF 2.0 architecture is provided in Figure 2 labelled as JMF 2.0 Architecture.

Among other alternatives to implement CGU client, JMF was selected since Java was decided to be used for client development. Further, JMF can be used in applets and in stand-alone applications developed in Java. JMF also offers an easy to use library and source code.

CGUsipClientv1.1 architecture is presented in Figure 3. There are two main Java packages – *cgusip.client* and *cgusip.utils* – that structure CGUsipClientv1.1. The *utils* package handles the existing instances of sip connections and calls. The client package has three main components: *gui*, *sip*, and *media*. The *gui* package handles all aspects of the client user interaction. The *sip* package handles the necessary interaction between the CGUsipClientv1.1 and the Dynamicsoft sip stack for creating and terminating sessions and initiating and receiving calls. The *media* package is in charge of making media connections using JMF and Dynamicsoft media libraries.

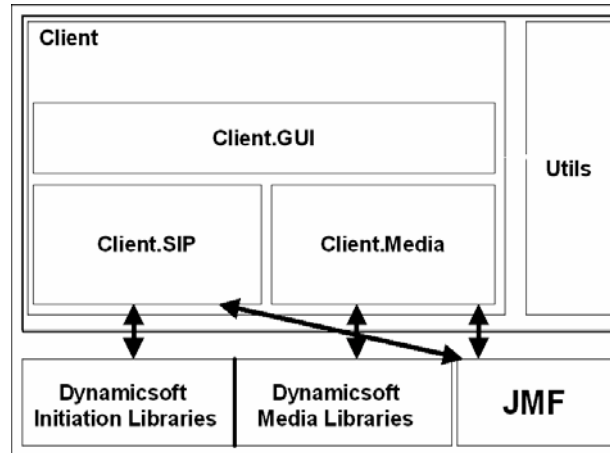


Fig. 3. CGUsipClientv1.1 Architecture

CGUsipClientv1.1 is capable of registering to a SIP proxy/registrar and making multiple calls (up to 5 lines are available) to any user capable of receiving SIP calls. In addition, the client provides connection to a private address book and the enterprise directory.

In order to download the installation files for CGUsipClientv1.1, each user needs to register through a website by providing minimum information about them: name, email address, and password for the sip client. During this registration process the following are created (1) a sip user account in the Oracle database, (2) an entry in the enterprise directory (ED) of the CGU NCL lab, (3) an entry linked to this ED user in the commObject, explained in section 3, (4) a private address book file in the web server. Figure 4 labelled as User Registration Process illustrates registration process.

CGUsipClientv1.1 also provides the callerID option for users. In the registration the user can upload a picture that will be used during the invite process to provide detailed information about the caller. Figure 4 labelled as Caller ID Process illustrates working of Caller ID feature.

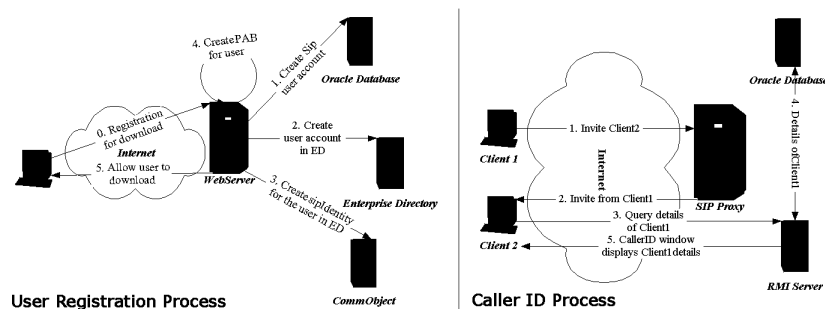


Fig. 4. User Registration and Caller ID Process Flows

Video formats that CGUsipClientv1.1 supports are h.263, h.261, and jpeg. H.263 is the default codec for video sessions and users are not allowed to change it. However, all three video formats are supported for incoming video. Future version of the client will allow users to change this parameter. The supported audio formats and their description are provided in Figure 5. The default audio codec is g.723 and this can be modified by the user

| Codec        | Appropriate Scenarios                                                                  |
|--------------|----------------------------------------------------------------------------------------|
| g.723<br>GSM | Best suited for PC-to-telephone and PC-to-PC calls. Used when the speed is 56K or less |
| DVI<br>ULAW  | Best suited for PC-to-PC calls. Do not use if the speed is less than 56K               |

Fig. 5. Supported Audio Codes

CGUsipClientv1.1 runs on local port 8000 and users are not allowed to modify this parameter. In the next version, this parameter will be configurable by users. The audio port range is from 4000 to 5000. The video port for this client is on 65500. We haven't tested multiple video performances on the same machine. Therefore, users are not allowed to change this port and receive video connections from different sources. In case of multiple calls, the video and audio of the active call is projected. Figure 6 shows a snapshot of a call between two clients.



Fig. 6. . CGUsipClientv1.1 snapshots

SIP and H.323 standards, which are used for video conferencing, do not include any solution for NAT and Firewall issues. Although some RFCs [8,9], various internet drafts [10,11,12,13] and industry practices [14] are being proposed to solve this problem, these have not been materialized into a standard yet. Therefore, CGUsipClientv1.1 does not work behind NAT or firewall. New version may provide NAT/FW support.

### 3 Middleware

Middleware refers to a suite of systems that exist between an application and various network services. Network services include providing platform for (a) security for authorization, authentication and secured transmission of messages (b) directories for identification and searching. Middleware tries to bind network services with the application. Middleware, in CGUsipClientv1.1, development focuses on implementing security and directory services.

For directory services, we have implemented an architecture that is recommended as a draft [15] by ViDeNet group of Internet2 community. One of the purposes of the group is to define a structure for video and voice communications that could reside in an Lightweight Directory Access Protocol (LDAP) directory. The draft proposed by the group suggests objects that need to be inherited by enterprises, which plan to implement directory structure for voice or video communications. The draft proposes creation of certain classes to support communication architecture. It suggests creation of two classes; commURIObject and commObject and other protocol specific classes that enterprise would choose to support such as SIP, H.323 or VRVS. Every enterprise that needs to support this structure has to create a protocol specific object to hold attributes for communication and update enterprise directory to maintain association between a user and their communication attributes. To achieve this, ED needs to be updated to include an attribute called commURI from commURIObject. commURI is an LDAP Uniform Resource Locator (URL) that refers to protocol specific attributes.

At CGU we have implemented commObject structure on openLDAP, an open-source directory service platform. In CGUsipClientv1.1 interface, there is a clickable icon that displays an html page onto a browser listing white page entries in ED. ED displays all users existing in that directory. Each entry in ED has a link, which is commURI 'pointer' as described above. commURI 'pointer' navigates to another page that enumerates various attributes that are needed for communication.

### 4 Experience with Deployment over Internet2

Network Convergence Lab (NCL) at CGU has made the software available to participants of ViDeNet<sup>2</sup> through its web site <http://ncl.cgu.edu/sipclient/index.php>. Through the web site, visitors intending to use the software have to provide their details including their email address. This email address becomes their unique SIPURI for registration purposes. A SIP proxy normally supports only one domain. However, since CGU SIP proxy is used by various campuses for proof of concept, multiple domain support was necessary. Therefore, domain part of each email address is verified against the domain database of the proxy. If a domain does not exist, the domain database is updated. In addition, every registered member is required to have an entry in ED. Further, every registered user is provided with a SIP URI on which they can be contacted if conferencing session is desired by other SIP UA. This SIP URI is one of the communication attributes in commURI. Other significant communication attributes being proxy domain address and registrar domain address. Proxy and registrar domain ad-

<sup>2</sup> Information on ViDeNet can be found at [www.vide.net](http://www.vide.net).

dress for every registered member is server address of Dynamicsoft proxy hosted by CGU in its campus. Every member who has downloaded the software registers itself with proxy at CGU. The website mentioned above provides an interface to directory services; ED and commObject attributes.

#### 4.1 Interoperability Issues

Before deploying on Internet2, CGUSipClientv1.1 had been successfully tested for point-to-point voice and video communication. There was a need to test the client with other clients and proxies after initial period of distribution.

NCL has two servers in its premises; a Linux-Based server hosting Vovida proxy and a Windows 2000 Advanced Server hosting Dynamicsoft (DS) proxy. For testing, a collaborative effort was arranged between Tim Poe at University of North Carolina, Chapel Hill, Tyler Johnson at University of North Carolina, Chapel Hill, and Chris Arnold at Radvision. Chris Arnold was using MSN Messenger and Siemens SIP client. Tyler Johnson was using MSN Messenger. Students at CGU were using CGUSipClientv1.1 application, Vovida's user agent and MSN Messenger. The results of interoperability test are provided in Figure 7.

| CALLER  | CALLEE  | Registered to  | Call Established | Audio Established | Video Established |
|---------|---------|----------------|------------------|-------------------|-------------------|
| cgu     | cgu     | DS,Vovida      | ✓                | ✓                 | ✓                 |
| cgu     | vovida  | Vovida         | ✓                | NT                | NA                |
| cgu     | msn     | DS,Vovida      | ✓                | ✓                 | X                 |
| cgu     | siemens | DS, Vovida     | X                | X                 | X                 |
| cgu     | mcu     | X*             | NT               | NT                | NT                |
| msn     | cgu     | DS, Vovida     | X                | X                 | X                 |
| msn     | msn     | DS, Vovida,RTC | ✓                | ✓                 | ✓                 |
| msn     | siemens | Vovida         | ✓                | ✓                 | ✓                 |
| msn     | siemens | DS             | ✓                | X                 | X                 |
| msn     | vovida  | Vovida         | ✓                | NT                | NA                |
| msn     | mcu     | RTC            | ✓                | ✓                 | ✓                 |
| siemens | cgu     | DS, Vovida     | X                | X                 | X                 |
| siemens | msn     | Vovida         | ✓                | ✓                 | ✓                 |
| siemens | msn     | DS             | ✓                | X                 | X                 |
| siemens | siemens | NT             | NT               | NT                | NT                |
| siemens | vovida  | NT             | NT               | NT                | NT                |
| siemens | mcu     | RTC            | ✓                | ✓                 | ✓                 |
| vovida  | cgu     | Vovida         | ✓                | NT                | NA                |
| vovida  | msn     | Vovida         | ✓                | NT                | NA                |
| vovida  | vovida  | Vovida         | ✓                | NT                | NA                |
| vovida  | siemens | NT             | NT               | NT                | NT                |
| vovida  | mcu     | X*             | NT               | NT                | NT                |

NT Not Tested

NA Not Applicable since vovida client does not have video support

(\*) They were never able to register to the same proxy

Fig. 7. Interoperability between different clients

To summarize the testing, it was found that all user agents except Vovida were able to register with Dynamicsoft proxy. MSN Messenger proved to be efficient in terms of interoperability performance as long as clients registered to Microsoft RTC server. MSN Messenger and Siemens SIP user agent were not able to call CGUSipClientv1.1. Media communication remain untested with Vovida user agent. Video

session could not be established between MSN Messenger and CGUsipClientv1.1 because MSN Messenger supports a proprietary video codec developed by Microsoft. Further, MSN Messenger, registered with Dynamicsoft, Microsoft RTC or Vovida, was not able to call CGUsipClientv1.1 because MSN Messenger does not provide for separate entries for Proxy and Registrar. Hence, a user cannot specify separate ports for Proxy and Registrar.

For conference testing Radvision MCU was used. Radvision MCU needs to register its services to a SIP proxy in order to provide conferencing services. During this registration it does not provide a SIP URI. It registers by three parameters, registrar, proxy, and domain. Once it registers its services, the users can call MCU by calling the assigned user name by the administrator. Due to this behavior, it cannot register to any proxy except Microsoft RTC server. As a result, any client who could not register to RTC was not able to join a conference call

#### 4.2 Performance Evaluation

During the preliminary performance testing of the CGUsipClientv1.1, two systems were used for a point-to-point video conferencing call. The configuration of these systems is provided in Figure 8.

Four metrics were identified for performance testing: CPU load, video frames per second, audio and video bit rates. Recent testing with CGUsipClientv1.1 provided the following performance results shown in Figure 8. All the values represent received video and audio performance ranges during a "2 minute" call.

The performance provided in Figure 8 is achieved after the initiation phase is over. During the initiation phase the CPU load changes as shown in Figure 9.

|                 | Configuration   |        |              |                  | Performance |                   |                          |                          |
|-----------------|-----------------|--------|--------------|------------------|-------------|-------------------|--------------------------|--------------------------|
|                 | CPU             | Memory | O/S          | Camera           | CPU Load    | Frames Per Second | Kbits Per Second (Audio) | Kbits Per Second (Video) |
| <b>System 1</b> | Pentium4 1.8GHz | 256MB  | Windows 2000 | Intel CS330      | 40-50%      | 10-17             | 63/5.3                   | 52.4 - 77.7              |
| <b>System 2</b> | Pentium4 1.8GHz | 256MB  | Windows XP   | Logitech Express | 40-50%      | 12-25             | 63/5.3                   | 65.5 - 120               |

Fig. 8. System configurations and their performances during the call or after the call is established

| Action                          | CPU load  |
|---------------------------------|-----------|
| Client was started              | 80%       |
| Registered to registrar         | 50%       |
| Call initiated                  | 30%       |
| Caller ID information requested | 45%       |
| Audio connection established    | 60%       |
| Video connection established    | 50% - 70% |

Fig. 9. Call initiation performance

## 5 Conclusion

In this paper, we highlight the following contributions:

- Built an effective SIP-based video conferencing desktop client that runs on top of SUN's JMF and a commercial SIP stack from Dynamicssoft
- We believe this is the first directory-service enabled video client that facilitates easy searching.
- The client provides authentication using native mode SIP authentication that uses the Digest mechanism with MD5 hashing.
- To make the download and deployment efficient, we have implemented web services that include support for commObject creation, user accounts on a location server, and caller ID service.

Our future work will involve analyzing interoperability problems encountered in greater details. We are planning implementation of an enterprise-wide authentication mechanism and exploring single-sign-on with Kerberos and X.509 digital certificates. We also intend to explore various authentication and authorization policies for video services. Finally a big challenge is to develop federated identity management and authorization in which various domains work with each other to obtain attributes about users and make decisions to forward a call or not. Those would be reported in a future article.

## Acknowledgment

We are indebted to several people who have helped shape our thinking that we have described in this paper. Everyone that makes Internet2 VidMid so special deserves our thanks. We thank Ken Klingenstein, Bob Morgan, Scott Canter and Michael Gettes for educating us with the federated administration concept. We also thank Egon Verharen, Tyler Johnson, Nadim El-Khoury, Tom Barton, Aditya Srinivasan, Doug Sicker, Jon Peterson and folks at RADVISION for several brain storming conference calls.

## References

- [1] R.S. Dixon, "Internet Videoconferencing: Coming to your Campus Soon!," *EDUCAUSE QUARTERLY*, no.4, Nov. 2000, pp.22-27.
- [2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol," *IETF RFC 3261*, June 2002.
- [3] International Telecommunication Union, "Packet based multimedia communications systems," *Recommendation H.323*, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Feb. 1998.
- [4] R. Radovic, I. Crkvenac; and S. Srbljic; "Formal definition of SIP end systems behavior," *International Conference on Trends in Communications EUROCON'2001*, vol: 2, pp. 293 - 296, 2001.
- [5] S. Salsano, L. Veltri, D. Papalilo, "SIP Security Issues: The SIP Authentication Procedure and Its Processing Load", *IEEE Network*, vol. 16, no. 6, pp.38-44, Nov/Dec 2002.

- [6] Dynamicssoft Proxy Server 5.2 Administrator's guide, 2001.
- [7] "Java Media Framework API Guide" [online] Mountain View, California 94043-1100 U.S.A., Nov. 1999 [cited March 11, 200], available from World Wide Web: <<http://java.sun.com/products/java-media/jmf/2.1.1/specdownload.html>>.
- [8] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, A. Rayhan, "Middlebox Communication architecture and framework" RFC 3303, Internet Engineering Task Force, August 2002.
- [9] R.P. Swale, P.A. Mart, P. Sijben, S. Brim, M. Shore "Middlebox Communications (MIDCOM) Protocol Requirements" RFC 3304, Internet Engineering Task Force, August 2002.
- [10] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy "STUN – Simple Traversal of UDP Through Network Address Translators" Version 05, Internet Engineering Task Force Internet-Draft, work in progress, Expires June 2003.
- [11] S. Sanjoy, P. Sollee, S. March "MIDCOM-unaware firewall/NAT Traversal" Version 01, Internet Engineering Task Force Internet-Draft, work in progress, Expires October 2002.
- [12] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, "Traversal Using Relay NAT (TURN)," Internet Engineering Task Force Internet-Draft, work in progress, Expires March 2002.
- [13] J. Rosenberg, J. Weinberger, H. Schulzrinne, "An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing," Internet Engineering Task Force Internet-Draft, work in progress, Expires March 2003.
- [14] "Network Convergence: An Overview of the Ridgeway IP Freedom Solution" Ridgeway Systems and Software, Austin TX, [cited March 11, 200], available from World Wide Web: <<http://www.wave3software.com/pdfs/ipfreedom/NetworkConvergence.pdf>>.
- [15] Thomas Barton, Nadim El-Khoury, Michael Gettes, Tyler Johnson, Sasha Ruditsky, Art Vandenberg, Egon Verharen: NSF Middleware Initiative Draft, work-in-progress, Expires November 2002; available from World Wide Web: <[http://middleware.internet2.edu/video/draftdocs/draft-nmi-edit-vidmid\\_vc-commObject\\_White\\_Paper-1.0.html](http://middleware.internet2.edu/video/draftdocs/draft-nmi-edit-vidmid_vc-commObject_White_Paper-1.0.html)>.