SIP-based Enterprise Converged Networks for Voice/Video over IP: Implementation and Evaluation of Components

Samir Chatterjee*, Member, IEEE

Bengisu Tulu, Member, IEEE

Tarun Abhichandani

Haiqing Li

Abstract— The next generation of enterprise networks is undergoing major changes as plethora of new architectures, applications and services begin to roll out within businesses. In general, the world of voice/telephony, video and data are "converging" into a global communications network. The purpose of this paper is two folds: First, the design, analysis and performance of a SIP-based videoconferencing desktop client, which has been developed and deployed over Internet2, is presented. Second, a guideline for managing SIP-based services to be deployed within enterprises, which addresses several challenges in each layer such as NAT/FW issues, directory service integration issues and interoperability issues, is proposed. Several detailed experimental results related to interoperability and conformance that were carried out are presented. Findings of extensive SIP/NAT traversal analysis through network traffic measurements are reported. The lessons learned from both the design of a new SIP based voice/video client as well as management challenges with enterprise deployment are highlighted.

Index Terms— VoIP, Videoconferencing, SIP, Architectures, Middleware, Security.

Manuscript received May 1, 2004; revised December 3, 2004. Portions of this work were also funded by a grant from the National Science Foundation (NMI 0222710). The authors are with the Network Convergence Laboratory at Claremont Graduate University, 130 E. Ninth St. Claremont, CA 91711 (corresponding author to provide phone: 909-607-4651; fax: 909-621-8546; e-mail: Samir.chatterjee@cgu.edu).

I. INTRODUCTION

THE next generation of enterprise networks is undergoing major changes as plethora of new architectures, applications and services begin to roll out within businesses. In general, the world of voice/telephony, video and data are "converging" into a global communications network. This paper deals with the technical and managerial aspects of implementing such converged network architecture and services. A large number of factors are involved in creating a robust enterprise network capable of delivering multimedia services. These factors include, but not limited to, better voice and video codecs, packetization, packet loss, packet delay, delay variation, directory services, resource integration and reliable network architecture. Also critical are the choices of call signaling protocols, security concerns, the ability to integrate seamlessly with existing Internet services and the need to traverse NAT and firewalls.

In this study, we focus on issues regarding the design, deployment and management of "converged" enterprise networks using the Session Initiation Protocol (SIP) [1] as the signaling platform. SIP, which is an Internet Engineering Task Force (IETF) standard for Internet Protocol (IP) Telephony, has received much attention recently and seems to be the most promising candidate as signaling protocol for the current and future IP telephony services, video services and integrated web and multimedia services. While SIP is new and actual deployment experiences are fewer, it is widely expected that future enterprise networks will incorporate SIP for its simplicity, flexibility, and built in security features. We note that H.323 [2] is also another signaling platform to build enterprise converged services. Instead of debating between the two protocols, we refer the readers to interesting literature on their comparison [3-5].

Although the evolution of the core enterprise network to IP is enabling the migration of the traditional circuit-switched voice and call signaling message traffic over the Internet using Voice over IP (VoIP) technology, there are many technical issues and challenges that need to be resolved for its successful commercial deployment. The purpose of our paper is to discuss those

issues and present existing solutions. However we first analyze the benefits offered by such a unified end-to-end IP-based multimedia network solution.

- *Cost reduction*: Moving voice calls over Internet eliminates the notion of long-distance. Further convergence of voice, data and video traffic can improve network efficiency and reduce operation cost.
- *Utilization*: Digitized voice calls require less bandwidth than the traditional 64 kb/s circuit calls and hence more calls can be made over the existing bandwidth.
- *Simpler Integration*: An integrated infrastructure allows more standardization and is simpler to manage. It is now possible to have tighter integration with web-based applications and supply-chains.
- *Enhanced Services*: Richer and enhanced services that integrates existing enterprise applications with VoIP, video or presence technologies is now possible.
- *Consolidation*: Since users are among the most significant cost elements in a network, any opportunity to combine operations, to eliminate points of failure, and to consolidate accounting systems, to track usage of resources, would be beneficial.

While enterprise customers clearly see the benefit of migrating to such converged networks, even the service providers have optimism to support such convergence.

- *More Revenue*: While traditional voice business is down, data is growing. Hence digitized voice and video services will provide them with more new revenue models.
- *Efficiency:* It has been proven that it is more efficient and cheaper to provision a packet-switched network than a circuit-switched network [6]. Hence the migration towards packet architecture is inevitable.
- *Ubiquitous Service*: The service providers will now be in a position to offer any service (voice, video or data) to any customer through their converged network.

The goal of this study is two-folds: first, to present the design, development and performance

of a SIP-based converged application; second, to provide a technical and managerial guideline for SIP-based enterprise deployments by highlighting issues identified throughout our experience with SIP-based systems. The rest of the paper is organized as follows: Section II gives a brief overview of the SIP protocol. Section III presents a technical guideline that addresses the important issues at each layer of the stack that one needs to consider before deploying any SIP-based enterprise architecture. Section IV presents the design, implementation, functionality and performance of SIP-based advanced desktop software that we have built for Internet2[1] and addresses the middleware support namely, the need for directory and security services. Section V presents experimental conformance and interoperability test results for SIP user agents (UAs). Section VI presents the NAT traversal issues and proposed solutions to tackle them. Section VII presents a management decision flow to guide decisions regarding SIP-based enterprise implementations. Section VIII summarizes the implications and lessons learned with SIP enterprise architecture and we conclude with potential future work in Section IX.

II. BRIEF OVERVIEW OF SIP

A brief overview of SIP is presented in this section. First, basic SIP call flow and SIP functionality is discussed. Later, the security mechanisms utilized by SIP are presented. Finally, a discussion on SIP packet structure is provided.

A SIP call flow

To make Internet multimedia (audio or video) calls, a caller must know the IP address and port number where the callee wants to receive audio/video packets as well as the audio and video codecs the callee supports. However, IP addresses are hard to remember and can easily change with users' mobility when they receive dynamic addresses through Dynamic Host Control Protocol (DHCP) servers. SIP facilitates user mobility by using high-level addresses of the form

[1] Internet2 (http://www.internet2.edu/) is a consortium being led by 207 universities working in partnership with industry and government to develop and deploy advanced network applications and technologies, accelerating the creation of tomorrow's Internet.

user@domain, which is called SIP URI (Uniform Resource Identifier). For instance, a user can call Alice at alice@abc.com regardless of what communication device, IP address, or phone number Alice uses. The high-level address is bound to the user's current location in SIP registrar servers, and the user's communication devices register with the registrar servers periodically by providing their current addresses (see Figure 1).

Figure 1 shows the steps involved when a user Bob wants to call another user Alice. Bob sends an INVITE message along with the session description protocol (SDP), carried in SIP requests and responses, which describes the list of supported audio and video codecs and the transport addresses to receive them. Once a call is established, Real Time Transport Protocol (RTP) and RTP Control Protocol (RTCP) are used to transfer media between Bob and Alice. A SIP Proxy server typically handles call routing, and redirect function can also be collocated with it.

INSERT > Figure 1: SIP call flow showing register and invite messages

B SIP security

The overall SIP protocol architecture from IETF is shown in Figure 2. It is important to protect the privacy of SIP users and guarantee confidentiality of their interaction. The mechanisms that provide security in SIP can be classified as end-to-end or hop-by-hop protection [3]. End-to-end mechanisms involve the caller and/or callee SIP user agents. SIP provides specific features (e.g., SIP digest authentication [7] and SIP message body encryption using S/MIME [8]) for these mechanisms. Hop-by-hop mechanisms secure the communication between two successive SIP entities in the path of signaling messages. SIP does not provide specific features for hop-by-hop protection and relies on network-level (IPSec) [9] or transport-level (TLS) [10] security. If a user address is expressed using a SIP Secure (SIPS) URI (sips:bob@biloxi.com), it means that the use of TLS is requested.

INSERT > Figure 2: IETF SIP Protocol adopted from [5]

SIP communications are susceptible to several types of attacks. They include *snooping,*

modification, spoofing, and denial-of-service[1, 7]. Such attacks make SIP enterprise systems vulnerable and hence it becomes even more important to design these networks with best possible security solutions.

C SIP packet structure

A SIP message is either a request from a client to a server, or a response from a server to a client [1]. The message consists of a start-line, message header, an empty line and an optional message body. Figure 3, below, illustrates the structure of the message and further details can be found in [1]:

INSERT > Figure 3: Structure of SIP Message

III. A GUIDELINE FOR CONVERGED ENTERPRISE SIP IMPLEMENTATION

We present a technical guideline in Figure 4 for a SIP-based converged enterprise architecture. For simplicity the well-known TCP/IP stack is also shown. For each stack layer, a set of important technical as well as managerial issues are listed. For the Application Layer, SIP supports various kinds of IP Telephony, Videoconferencing, Instant Messaging as well as web-integrated applications. Vendors have built a variety of IP hard-phones as well as soft-phones. Videoconferencing using SIP is still relatively immature which is the subject of our discussion in the next section. Instant Messaging using the SIMPLE [11, 12] standard is also maturing. The important issues that affect this layer includes design of SIP-based voice or video clients, the quality of media, overall performance and integration of converged applications with legacy enterprise software.

INSERT > Figure 4: A guideline for implementing SIP-based converged services for Enterprise

For the Transport Layer, it is best to describe it as a Middleware Layer for SIP. Besides the possible use of various transport layer protocols that can carry SIP packets [13], many other middleware services are needed. These include directories (white page lookup), provision for

security to ensure authentication and authorization, and solutions for the interoperability issues.

For the Network Layer, SIP utilizes TCP/IP. However, there are critical problems with NATs and firewalls [14, 15], implementing QoS and achieving interoperability. NATs and firewalls, behind which most enterprises sit, can disrupt SIP applications easily. For the Link Layer, SIP is oblivious since it is carried by IP protocol. The link can be wired (Ethernet) or wireless (IEEE 802.11b). However, the performance aspects of SIP applications and security issues over wireless are not well-studied. Similar studies for H.323 over wireless networks have been conducted [16]. Finally at the Physical Layer, it is important to design robust infrastructure that can withstand cyber attacks (e.g. DDoS).

IV. DESIGN, IMPLEMENTATION AND PERFORMANCE OF CGUSIPCLIENT

To provide a freely available, SIP-based Video/Voice over IP (VVoIP) tool for Internet2 members, we developed CGUsipClient v1.1.x (http://ncl.cgu.edu/), a java-based application implemented on Dynamicsoft SIP stack. It uses Java Media Framework (JMF) APIs for voice and video operations. CGUsipClient v1.1.x provides a number of functionalities as shown in Figure 5. Basic SIP functionality provides session setup and termination. Media functionality provides audio and video communication capabilities. CGUsipv1.1.x supports G.723, DVI, GSM and μ-law audio codecs and H.261, H.263 and JPEG video codecs. H.350 is an ITU-T standard developed through our work [17]. H.350 functionality provides an LDAP-based solution for providing directory information and single sign-on. Other features that CGUsipClient v1.1.x provides are redirection and caller-ID.

INSERT > Figure 5: Functional components of CGUsipV1.1.x

National Middleware Initiative (NMI - http://nsf-middleware.org/Middleware/) proposes middleware as a layer of software residing between network and traditional applications to offer services such as managing security, access and information exchange. The initiative provides

these services to enable effective, scalable and transparent usage of collaborative and communication tools. Adopting from this vision, ViDe (http://www.vide.net/), a Video Development Initiative from Internet2 group, has developed H.350 (http://middleware.internet2.edu/video/docs/H.350/). H.350 provides a directory services architecture for multimedia conferencing currently for SIP, H.323, H.320 and generic protocols.

H.350 is an LDAP object class specification designed to standardize the way multimedia conferencing information is stored and accessed. By maintaining an enterprise directory with user information as well as a communication directory for endpoint device and protocol information, H.350 simplifies configuration management and scalability in an enterprise. A detailed discussion on H.350 can be found in [17]. CGUsipClientv1.1.x is the first SIP client to utilize this directory structure to offer *"White Page"*, *"Click-to-Call"* and *"Single Sign-On"* facilities. *"White Page"* displays a list of users with SIP URIs who are in the enterprise directory. *"Click-to-Call"* enables a user to call another user by "clicking" on the other user's SIP URI. *"Single Sign-On"* provides facility of authenticating with a SIP proxy/registrar based on the credentials fetched from the LDAP structure instead of explicitly providing for the username and the password for registration. A snapshot of CGUsipClientv1.1.x is shown in Figure 6. More details on our client can be found in [18].

INSERT > Figure 6: Snapshot of CGUsipClientv1.1.x.

The performance of CGUsipClient was evaluated by making a point-to-point videoconferencing call between two systems. The configuration of these systems is provided in Table 1.

INSERT > TABLE 1. PERFORMANCE TEST CONFIGURATION

Four metrics were identified for performance testing: CPU load, video frames per second, audio and video bit rates. Recent testing with CGUSipClientv1.1.1 provided the following performance results shown in Table 2. All the values represent received video and audio

performance ranges during a “2 minute” call.

INSERT > TABLE 2. PERFORMANCE METRICS AFTER INITIATING THE CALL

The performance provided in Table 2 is achieved after the initiation phase is over. During the initiation phase the CPU load changes as shown in Table 3.

INSERT > TABLE 3. CALL INITIATION PERFORMANCE

V. EVALUATION: CONFORMANCE AND INTEROPERABILITY

Interoperability is an important issue for VVoIP because there are multiple protocols and devices that provide similar services [19]. Accordingly, multiple protocols and devices need to be examined to decide whether they provide services as prescribed by standards. In enterprise-wide VVoIP deployments, local administrators have full control over the systems and devices selected for implementation. However, the need to communicate with others outside the enterprise is crucial and local administrators do not have control over external domains. Therefore, administrators need to take into account the conformity of clients with the standards as well as interoperability between various clients while selecting solutions for their enterprise. Evaluating conformity will provide administrators the disparity of their implementation of SIP entities with the standards. The further away an enterprise is from the standards the lesser flexibility is available to it in terms of implementing better solutions that are offered by research community. Moreover, examining interoperability between SIP products, available for implementation, will indicate the limitations of an enterprise in communicating with external organizations engaged in VVoIP. This study examined various SIP clients, as illustrated in Table 4, for conformance and interoperability testing.

INSERT > TABLE 4 – CLIENTS TESTED

The first subsection presents the results of tests conducted through HCL Technologies Conformance Test Suite and Sip Test Tool [20]. HCL’s SIP Test Tool can be used for testing

conformance to standard, as well as load, regression and integration testing. It acts as a distinct SIP entity and responds to the test automation needs of SIP based products such as SIP user agent, conference servers, and proxy servers. Use of this test tool is limited to the basic conformance tests of user agents in this paper. The goal is to investigate the basic protocol implementation of the clients being tested.

The second subsection presents the interoperability test results between the selected SIP clients. Among the proposed SIP interoperability scenarios [21], we selected a scenario in which we can test the capability of clients to make a basic voice and video call, if possible, to another client through a single proxy. Using a single proxy reduces the complexity of the session initiation and helps us focus on the client side only.

A. Conformance

Methodology

A SIP-based user agent or an endpoint is "logical entity that can act as both a user agent client and user agent server" [1]. The conformance testing was conducted based on these two functionalities – User Agent Server (UAS) and User Agent Client (UAC). Another set of conformance tests was conducted that examined capability of an end-point to receive or transmit SIP calls within a dialog or outside a dialog. A dialog, as defined in [1], represents a peer-to-peer SIP relationship between two user agents that persists for some time.

Simple call testing was conducted to evaluate the basic call setup functionality of a client within a dialog and outside a dialog. The process of communicating SIP-based call between end-points is similar in both cases with one difference being that the communication within a dialog necessitates generation of "Tag" header from an end-point that responds to the very first request it receives. The mechanism of "Tag" parameter in combination with "Call-Id" is used to identify a dialog between two SIP end-points [1]. Simple call setup functionality is designed to test

whether an endpoint is capable of generating a request or response based on the type of message it receives. In this group of tests, every endpoint was used to make or receive a SIP-based call to or from an HCL server. There were three subtests conducted under this set of testing. Figure 7 illustrates a successful call attempt from "User Agent" to HCL server.

INSERT > Figure 7: Simple Call Test Scenario

UAS tests are designed to test the server behavior of an endpoint under various conditions. In this set of tests, a SIP-based end-point receives requests from the HCL server. The requests generated for the end-point is transmitted with errors in them like providing unknown header parameters, unknown method or supplying malformed requests. In all these behaviors, if the endpoint does not respond as prescribed by [1], it is considered to have failed that particular test. Overall, 17 different tests were performed under UAS testing and they can be grouped into three categories:

- **Method:** Evaluate behavior of an end-point when it receives an invalid method in the request. There was one subtest in this category.
- **Header:** Evaluate behavior of an end-point when it receives a malformed header in any of the header parameters or receives unknown parameter in the header. There were twelve subtests in this category.
- **Request:** Evaluate behavior of an end-point when it receives request with or without mandatory parameters in "Request-Line", shown in Figure 3. There were four subtests in this category.

Under UAC set of tests, a SIP-based end-point generates a request to the HCL server and the HCL server responds to the request. This response is generated with malformed or invalid response messages or that expected a specific type of information to be transmitted from the endpoint. Similar to UAS tests, if the endpoint does not respond as prescribed by RFC 3261, it is considered to have failed that particular test. Overall, twenty-four tests were performed under

UAC testing and they can be grouped into three broad categories:

- **Request:** Evaluate an end-point when it is expected to generate request with specific parameters. There were 3 tests in this category.
- **Responses:** Evaluate whether an end-point is able to handle different kinds of responses. There were 20 tests in this category.
- **Redirection:** Evaluate whether an end-point is capable to handle the redirection response from the other end-point. There was 1 test in this category.

Results

Table 5 presents a summary of overall results. The results of simple call test indicate that Helmsman SIP-based client was unable to receive or transmit a basic call. However, it was able to transmit messages based on a dialog.

INSERT > TABLE 5 – SUMMARY OF CONFORMANCE TEST RESULTS

In UAS testing, it was observed that Messenger™ was capable of responding appropriately in almost all cases except where the header in the requests were malformed or had unknown values in header fields. A specific pattern of behavior stayed undetermined for Siemens client. While it was able to respond appropriately to certain requests, it failed to serve other requests that were malformed or incorrect. Grandstream and sipc were almost identical in their performance – both of them unable to detect the errors in headers of the message. CGUsip, Cisco and Session™ displayed variable behavior. While all of them could not trace the errors in header fields, CGUsip and Session™ detected errors in methods supplied whereas Cisco was able to respond appropriately when the requests had errors in them. Helmsman was unable to detect errors in almost all of the messages.

In UAC testing, CGUsip was unable to respond appropriately to certain response codes communicated to it by the HCL server. Messenger™, in addition to not responding to response codes, was unable to detect malformed responses or serve multiple responses for a request and

was unable to serve request for redirection. Session™ was able to trace malformed responses but was not able to respond to certain response codes. Similar was the case for Siemens but with certain variations. Although the number of tests passed by Cisco and Grandstream was same, the types of tests were different. Grandstream was able to trace fields in header whereas Cisco was able to serve various response codes. The results for sipc were similar to Grandstream, however, sipc was not able to parse certain header fields. Helmsman was not able to parse response codes, header fields or multiple requests or responses.

B. Interoperability

Methodology

This group of tests was based on a single proxy scenario presented in Figure 8. All selected user agents were registered to a single proxy and a testing protocol was established to test (S) Establishing a session, (A) Establishing audio communication, (V) Establishing video communication (if possible) between two different clients. Evaluating the quality of media is done in a subjective manner.

INSERT > Figure 8: Interoperability Testbed

Results

The results of the interoperability tests are presented in Table 6. The highlighted columns indicate complete interoperability between two clients within their respective voice or video capabilities. For example, in Table 6, when a call is initiated from Messenger™ to sipc the clients were able to establish the session, audio as well as video – indicating complete interoperability between them, however, this is not the case between Messenger™ and CGUsip. Only the calls between different clients are reported in the table. The results of Client A making a call to Client B is different in some cases than Client B making a call to Client A. This is caused by the UAC and UAS implementation differences in handling events.

INSERT > TABLE 6 – INTEROPERABILITY RESULTS[2]

Three clients in the testbed (Cisco, Grandstream, and Helmsman) are audio only clients and hence no video connection was expected from them. Moreover, even though Session™ is a video client since it only supports a proprietary video codec, it was not possible to make video communication between Session™ and any other client.

Cisco phone was able to communicate with all the clients except Helmsman. They failed to establish an acceptable audio communication since the audio received by Cisco was not understandable where as the audio sent by Cisco was perfect on the Helmsman side.

Grandstream hard phone passed all the tests with one exceptional case of Grandstream calling Session™. The implementation at Session™ for codec negotiation is the main reason for this failure. If only a single codec is selected in Session™ side, this problem does not occur and the audio communication can be established.

Session™'s codec negotiation problem is related to the way the codec list is presented in 200 OK messages. The Session Description Protocol (SDP) [22] states "When a list of payload formats is given, this implies that all of these formats may be used in the session, but the first of these formats is the default format for the session." Session™ responds back with a list of codecs that it supports without changing the order according to the incoming codec list in the INVITE message. This results in audio communication failure if the two lists are not in the same order since it misleads both clients and ends up with the exchange of audio using different audio codecs. Among video clients, Messenger™ and Siemens were the only two clients that established a session in any direction and shared both audio and video in any order.

Implications derived from evaluation

The experiments presented in this section are based on certain predefined scenarios that are common in real life implementations. However, the same measurements in a field study

[2] Y and N stand for Yes and No respectively. NA stands for Not Applicable.

environment can provide different results. Therefore, further studies needs to be conducted to better understand interoperability and conformance issues in an enterprise setting. It is also important to evaluate the extra features provided by clients in terms of how these features are utilized and their effects on the basic functionality, network load, and other applications. It is also important to note that during our experiments, quality of audio and video was measured subjectively. Further studies can be carried out that can evaluate the audio/video quality of the clients by using both objective and subjective measurements.

VI. SIP Over Network Address Translator (NAT)/Firewall Traversal

The challenge of traversing NAT and/or firewalls is still a barrier for VVoIP deployments [14]. NAT and/or firewalls pose two challenges for these technologies. Firstly, if NAT is placed between a SIP-based user agent (UA) and the Internet, the UA is allotted a private network address, which is not valid in the Internet. As a result, contact information (IP address and port number) is invalid for the external networks and responses from external proxy servers may not reach the UA. Another challenge is related to media sessions. During the SIP session initiation, RTP and RTCP ports are negotiated for establishing a media session between two user agents [1]. Even if the negotiation is successful, NAT or firewall will disallow the direct connection using the ports negotiated.

We have tested the performance of some existing NAT/Firewall solutions using network traffic analysis techniques. We have analyzed and evaluated three popular solutions for SIP over NAT traversal, which are: (1) an IETF Standard called Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (STUN) [23], (2) Universal Plug and Play (UPnP) [24], and (3) a proprietary solution by Ridgeway Systems Inc. - IPFreedom™ [25]. Even though other solutions have been proposed by IETF such as r-port [26] and MIDCOM [27], the common deployment of these three solutions motivated the choice of selection. Another recent

solution, Interactive Connectivity Establishment (ICE) [28] was not tested as we did not have any open source implementation available.

A. Brief Overview of STUN, UPnP, and IPFreedom™ Solutions

STUN [23] is used to resolve whether a UA is behind a NAT and if it is, the type of that NAT. Implementing a solution for STUN requires a STUN server external to the network that is being protected behind a NAT and a STUN-enabled client on a SIP device. A STUN-enabled UA requests external IP and port that it can use to form SIP headers when it initiates a session with another UA placed external to the network. These ports are related to SIP signaling as well as media.

UPnP [24], targeted at small-business users and residential installations [29], has been popularized by Microsoft. It is an extension of Device Plug and Play (PnP) providing a solution for traversing NAT for communication with external networks. It includes the entire network, enabling discovery and control of devices, including networked devices and services, such as network-attached printers, Internet gateways, and consumer electronics equipment [24]. This solution, unlike STUN, does not require a server outside the network but it requires a UPnP-enabled NAT device and a UPnP-enabled SIP UA.

IPFreedom™ [25] is a solution proposed by Ridgeway Systems for traversing VVoIP over NAT and firewalls. This solution works for all types of NATs and firewalls. It does not require configuration modifications on firewall or a NAT device. In this solution, a UA in private network establishes outbound communication connections through the NAT with a Ridgeway server on the public network. Signaling messages are transmitted through the server via Transmission Control Protocol (TCP) tunneling and media packets are transmitted over User Datagram Protocol (UDP) connections. Further, IPFreedom™ solution requires colocation of an IPFreedom™ client and SIP UA. The user needs to configure the SIP UA to use the

IPFreedom™ client. This solution can be used with various UAs without any modifications.

B. Experiment Methodology

For tracing different behaviors of three solutions, we compared benchmark (No NAT/Fw) and experimental scenarios. For both scenarios, Vocal, an open source SIP proxy by Vovida (http://www.vovida.org), was utilized. Further, Ethereal (http://www.ethereal.com) was placed in every network in the study to gather traffic on the network and trace the behavior of the packets being transmitted on the network.

The benchmark scenarios were categorized based on different clients that were used for the study that provided solutions for NAT traversal. As per Figure 9, in Grandstream Benchmark scenario, Grandstream BudgeTone SIP endpoints were used. In Microsoft XP Messenger™ Benchmark scenario, Windows Messenger™ 4.7 on Windows XP operating system clients was used. In Wave3 Session Benchmark scenario, Wave3 Session 2.1.5 clients were used. These clients and the Vovida proxy/registrar were placed on a single public network. Experiments in benchmark scenario involved establishing and terminating calls, making audio and video calls between the clients using Vovida proxy. The traffic measurement involved counting the number of messages transmitted for video and audio calls and tracing the process delays.

INSERT > Figure 9: Benchmark Scenario Design

A small-business scenario was considered for the experimental setup. This scenario involved placing a client in a network behind a NAT making calls to clients in the public IP network through a proxy or registrar in the public network. The scenarios in Figure 10 are categorized based on the solutions for NAT traversal, explained before.

INSERT > Figure 10: Small-Business Experimental Scenario Design

In STUN scenario, a Grandstream BudgeTone SIP Phone that provides STUN capability was placed behind a NAT and another on the public network. In the UPnP scenario, Microsoft's

Windows XP Messenger™ was used on internal as well as external network. In IPFreedom™ scenario, Wave3 Session™ client was used to test IPFreedom™ solution.

C. Performance Results

Results collected from Ethereal for each experiment were analyzed: (1) to measure the load on the network infrastructure by measuring SIP message traffic on the network, and (2) to calculate process delays caused by NAT traversal solutions. Three basic processes were evaluated – REGISTER message with the SIP proxy/registrar, INVITE and BYE messages between SIP UA. The analysis of message count, based on the exchanged messages between participating SIP entities was done for the three solutions. A summary of the results is presented in Table 7 for Message Counts.

INSERT > TABLE 7 – SUMMARY OF MESSAGE COUNTS

Table 7 is organized based on different processes and solutions for NAT traversal. For INVITE and BYE messages the experimental scenario is classified as "Pub>Prv", indicating calls made from client on the public address to clients on the private address, and "Prv>Pub", indicating calls made from client on the private network address to clients on the public network address.

As per Table 7, STUN solution resulted in increase in the number of messages in processes. The UA on the private network uses four additional STUN messages compared to the benchmark scenario. The four additional messages are related to STUN request and response. Similarly, for INVITE, there were two additional messages exchanged between the clients. There were no changes in the process of call termination. For UPnP, the process of UA negotiating IP and port with NAT/Firewall results in dramatic increase in the number of messages (85 in Table 7) for registering with the proxy/registrar. There were similar increases in INVITE as well as BYE messages. For IPFreedom™, the increases in the message counts were not as the same magnitude as UPnP but the solution did result in a small increase of messages exchanged

between SIP entities.

Table 8 illustrates summary of the delay occurred during the experiments in the registration and call processes. For calculating process delays, ten calls were made between clients. Average (Avg) of these calls were considered after ignoring the highest and the lowest value for each process. Standard deviations (Stdv) of these calls were calculated to indicate the variability of the delays in the calls.

INSERT > TABLE 8 – PROCESS DELAY FOR EACH EXPERIMENT (SECONDS)

For measuring registration delay, registration interval was calculated starting from the time REGISTER request was sent by the client to the time 200-OK message was sent by the server. Invite interval was calculated starting from the INVITE request was sent to the time 180-RINGING message was sent by the server back to the client.

There were not any marked delays for the processes in STUN solution when compared to benchmark. There was a 0.01 second difference in performance for INVITE message when a UA in public network initiated a call to a UA in private network. Another noticeable difference was apparent when a UA in private network initiated a call to a UA in public network. However, neither of these was significant compared to other implementations. The process delays in UPnP were noticeably higher compared to STUN. For example, the shortest delay observed was 2.54 seconds compared to the longest delay of 0.7 second in STUN scenario. Performance of IPFreedom^{TM} solution in general showed higher values of delay compared to STUN and UpnP. The delay reduced when INVITE messages were transmitted between clients in private network and in public network. However, there were no delays in the REGISTER process.

The results indicate that STUN solution was the most efficient one in terms of traffic load, message count, as well as the process delay and causes minimal performance impact on existing networks. However, it cannot solve the traversal issue for all NAT types and firewalls. UPnP generates a large amount of TCP messages that affect the network load behind the NAT. This

may not be problematic for a residential user using UPnP, whereas for an enterprise network, managers can expect unnecessary traffic that can affect the performance of other applications. It is important to note that this traffic is never reflected to the public network and UPnP can solve a large number of traversal problems. On the other hand, due to the fact that enterprises have a lot of Intranet activity that can get hampered, it may not be suitable for an organization where a large number of VoIP traffic is expected everyday. Finally, IPFreedom™ is a relay solution where the communication completely depends on a server outside, and hence results in higher delays. It causes increased TCP traffic in both private and public network. More TCP traffic will consume bandwidth on both public and private side, taking away available bandwidth for real-time applications. Even though we did not measure the media delay caused by the relay technique, it is expected to be higher compared to other solutions.

Length of INVITE process is very important since it represents the waiting time for users to establish a call. During the INVITE process for audio-only calls, IPFreedom™ solution caused higher delays in public as well as in private network. On the other hand, UPnP solution caused higher delays in private network for INVITE process for audio-video calls. This implies that depending on the use of audio and video, enterprise networks will experience delays both in public and private sides of the network with these two solutions.

VII. MANAGERIAL IMPLICATIONS

Previous sections provide a guideline for technical issues enterprises have to face while deploying SIP-based systems. This section provides a managerial guideline, illustrated in Figure 11, which can help managerial decision makers while identifying the need and finding solutions for enterprise implementations.

INSERT> Figure 11: Management Decision Flow for SIP-based Implementations

Every implementation requires a preparation and a decision-making phase. The preparation

phase should start with making a business case for deploying VVoIP systems in the enterprise. Expected returns for stakeholders should be identified to support the business case. Once a clear business case is identified, the management should first initiate an audit of existing infrastructure of the enterprise. Outcome of this audit should be a detailed list of enterprise capabilities. Identifying needs based on the existing capabilities of the enterprise infrastructure and the business case made in the first step should follow next. Requirements for the project will be the final outcome of this step. Identifying capabilities and requirements completes the preparation phase and initiates the decision-making phase.

Decision-making phase starts with a request for proposal where capabilities and requirements are listed with the objectives of the project and the expected timeline. Solution analysis starts after the proposals are received. During this step it is important to consider both technical issues, such as interoperability, conformance, and SIP/NAT traversal, as well as managerial implications. Depending on the objectives and the scope of the project, different SIP entities may be necessary for deployment. We categorized them as clients and servers. Distinction should be made between these two while doing solution analysis since their functionalities vary. Once a decision is made for a specific set of solutions, a pilot implementation is usually very useful to evaluate the selected solution. Full-scale implementation follows successful pilot implementations and if problems occur during the pilot implementation new solutions can be selected for testing. Once a full-scale implementation is in place, management should analyze the match between the results of the current deployment and the business case made at the preparation phase.

VIII. SUMMARY: IMPLICATIONS AND LESSONS LEARNED

As mentioned above, the client developed in our lab was based on a commercial SIP stack provided by Dynamicsoft. Along with other commercial stacks, there are a number of open

source stacks available for development such as Vocal, JAIN (https://jain-sip.dev.java.net/), OSIP (http://www.gnu.org/software/osip/osip.html) and SIP Express Router (http://www.iptel.org/ser/). These options need to be evaluated for RFC 3261 [1] compatibility, without which the development activities could get hampered in the long run. The future plan of our lab is to migrate from a commercial stack to an open source SIP stack. However, we do recognize that open source stack might have a limitation of not being fully updated with evolving standards.

Further, we have used Java as our development platform. There are some disadvantages of using JMF. We have experienced high CPU loads and memory usage. The time to start a video process is also longer.

Results of conformance testing indicated that some SIP UAs used in our experiments do not adhere to RFC 3261 fully. This might create performance problems in the long run as the decisions by administrators to opt for clients that do not interoperate with similar systems might not be well received by the users in the organization.

NAT/Firewall traversal is an important issue in enterprise deployments. As yet, there has not been a single comprehensive solution for overcoming this difficulty. This solution is of paramount importance for security and scalability of enterprise networks. Through our analysis, we have found that STUN [23] is effective but does not provide a solution for all types of NAT or firewall deployments. UPnP [24] is a popular solution for small office/home office networks. However, it forces serious delays in the processing of messages and increases delay significantly. IPFreedom™ is a proprietary solution that could work with most types of NATs and firewalls. Nonetheless, it generates traffic on the network due to TCP tunneling between entities. Enterprise networks need an efficient solution that requires minimum message exchange and do not generate significant impact on the performance of end-user multimedia applications.

Currently, configuring different SIP clients is still problematic. In this study we have tested

eight SIP user agents including commercial and freeware applications. The configurations of these clients vary among themselves and lack standardization. Configuration of these different clients, if not standardized, may turn into an administrative nightmare. An ideal option would be to devise a centralized solution for configuring clients in an enterprise network. A centralized solution will minimize advanced administrative issues like specifying the availability or unavailability of users at a given date/time or indicating the mode of communication like voice, video or messaging at a given date/time. H.350, an ITU-T standard, could be a potential solution for this problem by providing a standard architecture for SIP as well as other protocols for storing configuration information and user authentication information inside LDAP directories. H.350 could also provide central control interface for device management and user management.

IX. Conclusion

This paper has presented a summary of our work based on substantial experience in dealing with SIP-based multimedia services deployment within enterprise. We have described the design and architecture of a new SIP-based video client while highlighting the lessons learned from the process. We have also raised important deployment issues in various stack levels, presented the experimental analysis of the solutions that are available and lessons learned. We hope that the technical and managerial guidelines provided can be useful to network administrators and managers who are contemplating on deploying SIP-based solutions. In this paper, we did not focus much on securing SIP communication. As a future work, we see a lot of challenges in completely securing this SIP environment. The mechanisms of using digital certificates, certificate authorities (CAs), S/MIME for end-to-end encryption and managing trusts between domain CAs will be critical as we move towards a more robust production environment.

Acknowledgments

We would like to acknowledge the help from several people during the course of this project.

In particular we would like to thank Jeff Fildey of Wave3 Inc., Bill Zhang of Grandstream Networks Inc., and Pathangi N. Janardhanan of HCL Technologies for providing us with their software and tools. We also acknowledge the contributions of Jill Gemmill, Tyler M. Johnson, Egon Verharen and Nadim El-Khoury. Thanks to VidMid-VC working group within Internet2 for email discussions.

REFERENCES

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, N. Handley, and E. Schooler, "SIP: Session Inititation Protocol," Internet Engineering Task Force RFC 3261, June 2002.

[2] International Telecommunications Union, "Packet based multimedia communications systems," International Telecommunications Union, Recommendation H.323, November 2000.

[3] S. Salsano, L. Veltri, and D. Papalilo, "SIP Security Issues: The SIP Authentication Procedure and its Processing Load," *IEEE Network*, vol. 16, no. 6, pp. 38-44, 2002.

[4] Packetizer, 2003, "Comparisons between H.323 and SIP," http://www.packetizer.com/iptel/h323_vs_sip/, *accessed on: April 29, 2004*.

[5] J. Glassman, W. Kellerer, and H. Muller, "Service Architectures in H.323 and SIP: A Comparison," *IEEE Communications Surveys*, vol. 5 (2), pp. 32-47, 2003.

[6] S. Dunstan, "Converged Voice and Data Services in the New Network," *Telecommunications Journal of Australia*, vol. 51, no. 2, pp. 17-31, 2001.

[7] K. Ono and S. Tachimoto, "SIP Signaling security for end-to-end communication," in the Proceedings of The 9th IEEE Asia-Pacific Conference on Communications, Penang, Malaysia, 2003.

[8] E. B. Ramsdell, "S/MIME version 3 message specification," IETF RFC 2633, June 1999.

[9] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," IETF RFC 2401, Nov 1998.

[10] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," IETF RFC 2246, Jan 1999.

[11] M. Day, S. Aggarwal, G. Mohr, and J. Vincent, "Instant Messaging / Presence Protocol Requirements," Internet Engineering Task Force RFC 2779, September 2000.

[12] M. Day, J. Rosenberg, and H. Sugano, "A Model for Presence and Instant Messaging," Internet Engineering Task Force RFC 2778, February 2000.

[13] G. Camarillio, R. Kantola, and H. Schulzrinne, "Evalutaion of Transport Protocols for the Session Initiation Protocol," *IEEE Network*, no. September/October, pp. 40-46, 2003.

[14] V. Paulsamy and S. Chatterjee, "Network Convergence and NAT/Firewall Problems," in the Proceedings of The 36th IEEE Hawaii Conference on System Sciences, Hawaii, USA, 2003.

[15] M. Stukas and D. C. Sicker, "An Evaluation of VoIP Traversal of Firewalls and NATs within an Enterprise Environment," *Information Systems Frontier Journal, Kluwer Press*, vol. 6, no. 3, pp. 219-228, 2004.

[16] Sajal Das, Enoch Lee, Kalyan Basu, and Sanjoy Sen, "Performance Optimization of VoIP Calls over Wireless Links Using H.323 Protocol," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 742-752, 2003.

[17] J. Gemmill, A. Srinivasan, J. Lynn, S. Chatterjee, B. Tulu, and T. Abhichandani, "Middleware for Scalable Real-time Multimedia Communications Cyberinfrastructure," *Journal of Internet Technology (Forthcoming)*, vol. 5, no. 4, pp. 405-420, 2004.

[18] B. Tulu, T. Abhichandani, S. Chatterjee, and H. Li, "Design and Development of a SIP-Based Video Conferencing Application," in the Proceedings of 6th IEEE International Conference on High Speed Networks and Multimedia Communications HSNMC'03, Estoril, Portugal, 2003.

[19] G. W. Bond, E. Cheung, K. H. Purdy, P. Zave, and J. C. Ramming, "An Open Architecture for Next-Generation Telecommunication Services," *ACM Transactions on Internet Technology*, vol. 4, no. 1, 2004.

[20] HCL Technologies, "VoIP Product Testing: Challenges and Solutions," http://www.hcltech.com/voip/pdf/VoIP%20Testing%20White%20Paper.pdf, *accessed on: May 07, 2004*.

[21] International Multimedia Teleconferencing Consortium - SIP SIG Activity Working Group, June 3, 2000, "SIP Interoperability Scenarios Test Plan," http://www.cs.columbia.edu/sip/drafts/sipsig_interop.doc, *accessed on: May 07, 2004*.

[22] M. Handley and V. Jacobson, "SDP: Session Description Protocol - RFC 2327," Internet Engineering Task Force April 1998.

[23] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NAT)," Internet Engineering Task Force - RFC 3289 2003.

[24] Microsoft, 2000, "Understanding Universal Plug and Play - White Paper," http://www.upnp.org/download/UPNP_UnderstandingUPNP.doc, *accessed on: May 07, 2004*.

[25] Ridgeway Systems, 2003, "White Papers and Briefs," http://www.ridgewaysystems.com/support/whitepapers.aspx, *accessed on: May 07, 2004*.

[26] J. Rosenberg and H. Schulzrinne, "An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing - RFC 3581," Internet Engineering Task Force August 2003.

[27] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, "Middlebox Communication Architecture and Framework - RFC 3303," Internet Engineering Task Force 2002.

[28] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Multimedia Session Establishment Protocols," Internet Engineering Task Force (IETF), Internet-Draft draft-ietf-mmusic-ice-04, Expires: August 22, 2005, February 21 2005.

[29] Newport Networks, 2003, "Solving the Firewall and NAT Traversal Issues for Multimedia over IP Services - White Paper," http://www.newport-networks.com/whitepapers/, *accessed on: May 7, 2004*.

Figures

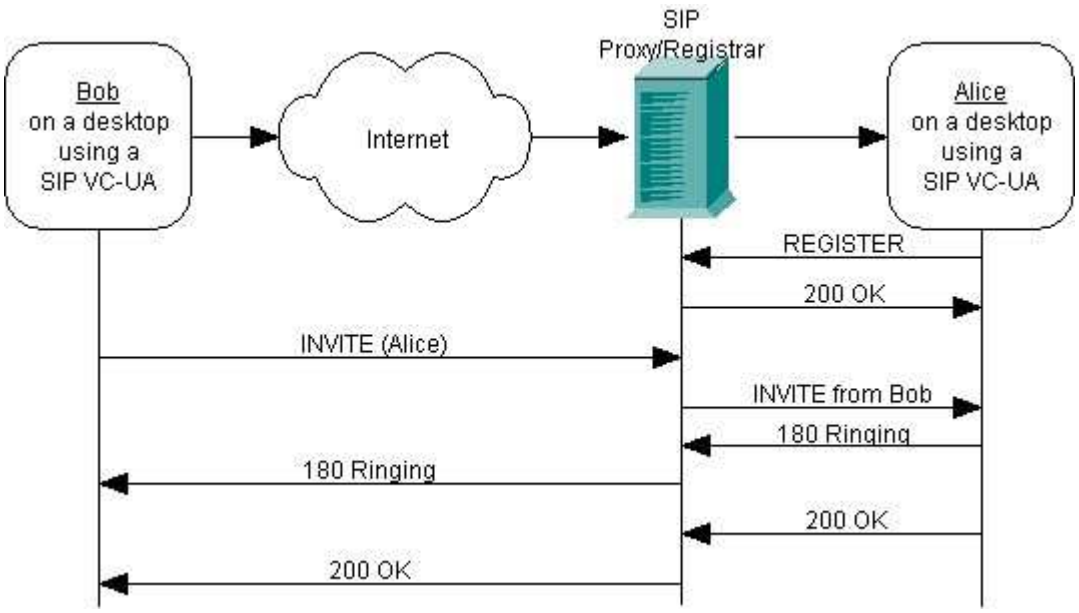

Figure 1: SIP call flow showing register and invite messages

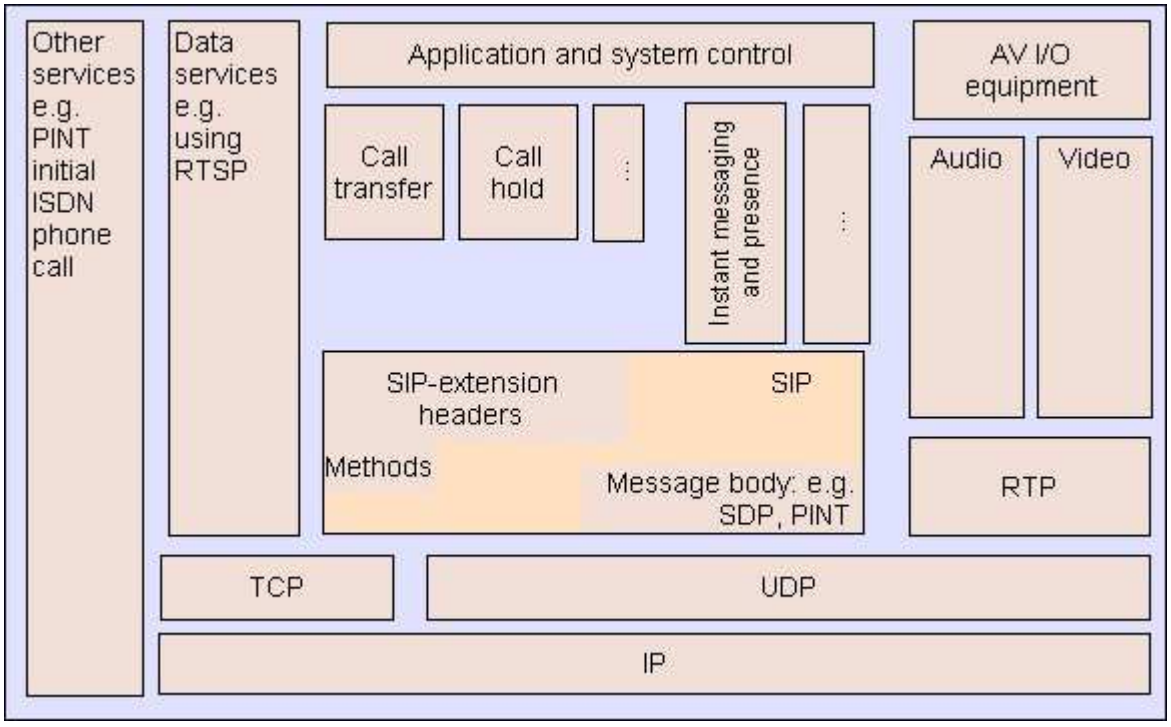

Figure 2: IETF SIP Protocol adopted from [5]

<table>
<tr><th colspan="4">SIP Message</th></tr>
<tr><th>Start-Line</th><th>Message Header</th><th>CRLF*</th><th>Optional Message Body</th></tr>
<tr><td>Request-Line</td><td>Name and Value</td><td rowspan="4"></td><td rowspan="4">- Session Description Protocol (SDP) message

- Subject: "Lunch"</td></tr>
<tr><td>- Method
- Request-URI
- SIP-Version</td><td><u>Format:</u>
Header : <Value> [,Value1,...]</td></tr>
<tr><td>Status-Line</td><td rowspan="2">- Via
- Route
- Record-Route
- Proxy-Require
- Max-Forwards
- Proxy-Authorization ...</td></tr>
<tr><td>- SIP-Version
- Status-Code
- Reason-Phrase</td></tr>
</table>

* - **Carriage-Return Line-Feed**

Figure 3: Structure of SIP Message

TCP/IP Stack	SIP-based Converged Architecture	Issues
Application	Advanced Converged Applications	- Design of new VoIP, videoconferencing and IM clients - Quality of media - Integration with web enterprise applications - Performance evaluation
Transport	Middleware	- Suitable transport protocol (RTP, SRTP, SCTP, RTSP, etc.) - Directories - SIP security - hop-by-hop, ETE, authentication, authorization - Interoperability
Network	TCP/IP	- NAT/Firewall - QoS - Interoperability
Link	Link (Wired, Wireless)	- Performance optimization over wireless - Security over wireless
Physical	Physical	- Robustness from cyberattacks

Figure 4: A guideline for implementing SIP-based converged services for Enterprise

Java-based CGUsipClient v1.1.x				
Functions	Basic SIP Functionality	Media ◆ Audio: u-law, g.723, DVI, GSM ◆ Video: H.261, H.263, JPEG	H.350 ◆ White Page Lookup ◆ Click-to-Call ◆ Single Sign-On	Other Features ◆ Redirection ◆ Caller-ID
Technologies	dynamicsoft SIP Stack Java - Abstract Window Toolkit Java - Swing	dynamicsoft SIP Stack Java - Abstract Window Toolkit Java - Swing Java - JMF API	openLDAP Java - Abstract Window Toolkit Java - Swing Java - JNDI	RMI HTTP Java - Abstract Window Toolkit Java - Swing

Figure 5: Functional components of CGUsipV1.1.x

Figure 6: Snapshot of CGUsipClientv1.1.x.

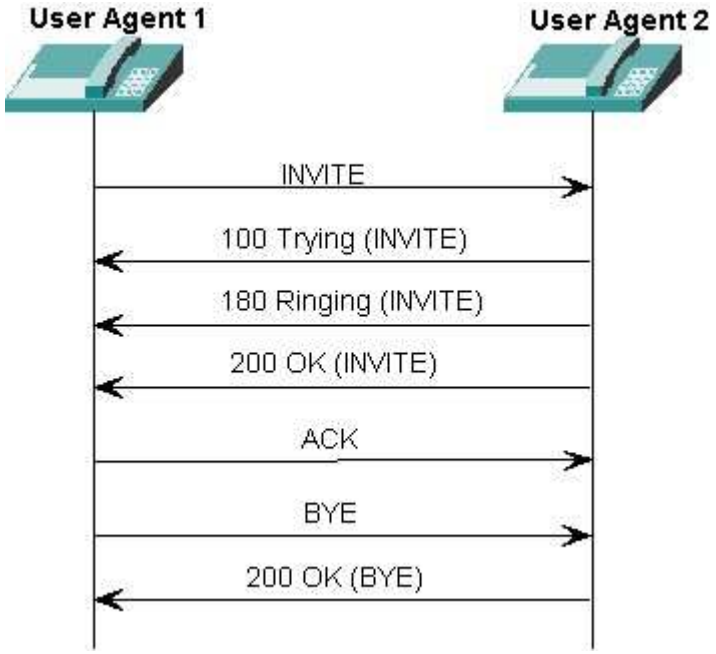

Figure 7: Simple Call Test Scenario

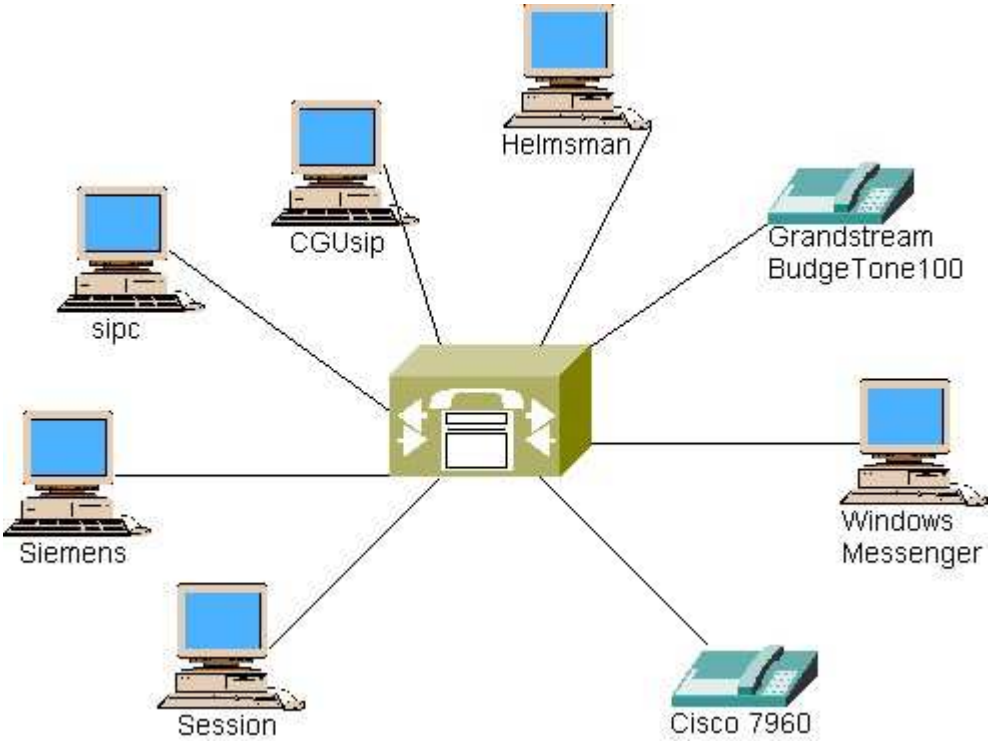

Figure 8: Interoperability Test Bed

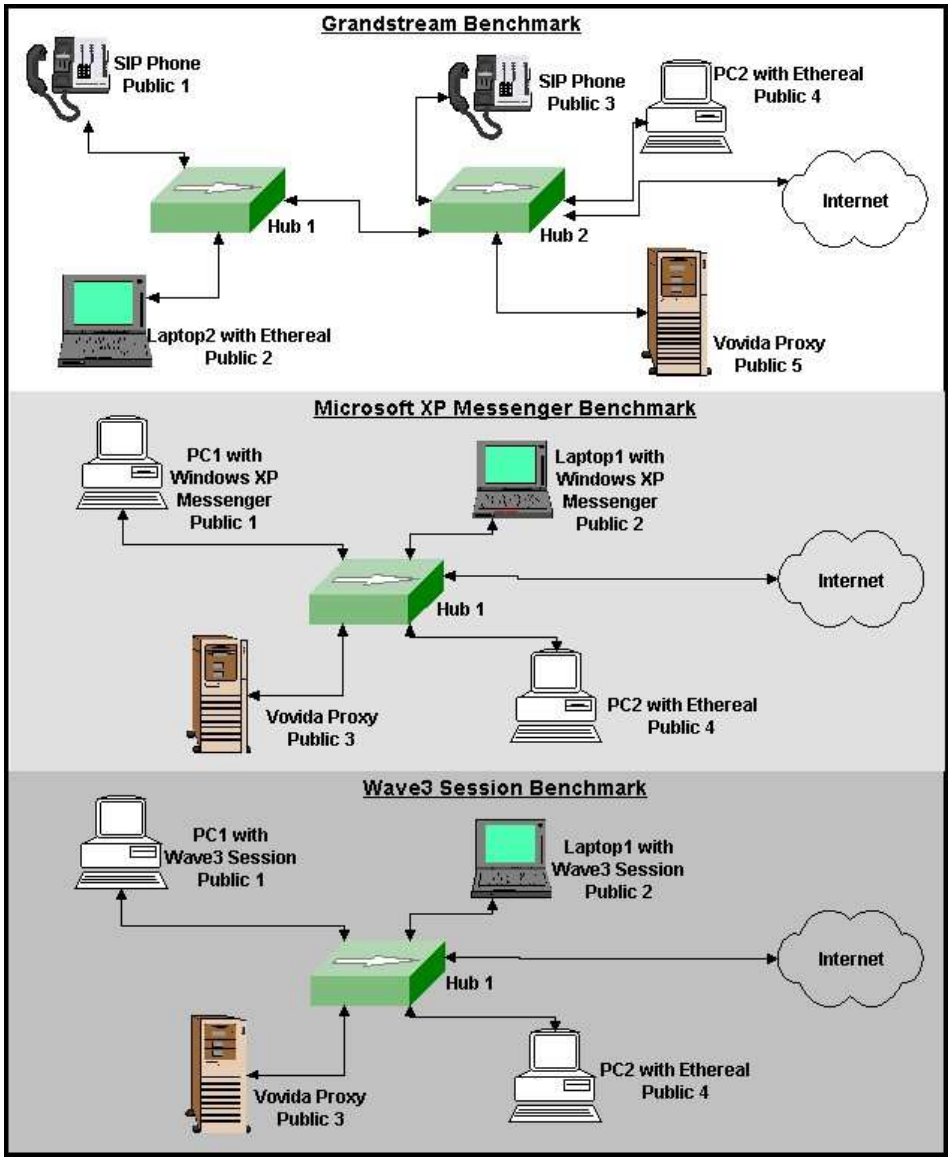

Figure 9: Benchmark Scenario Design

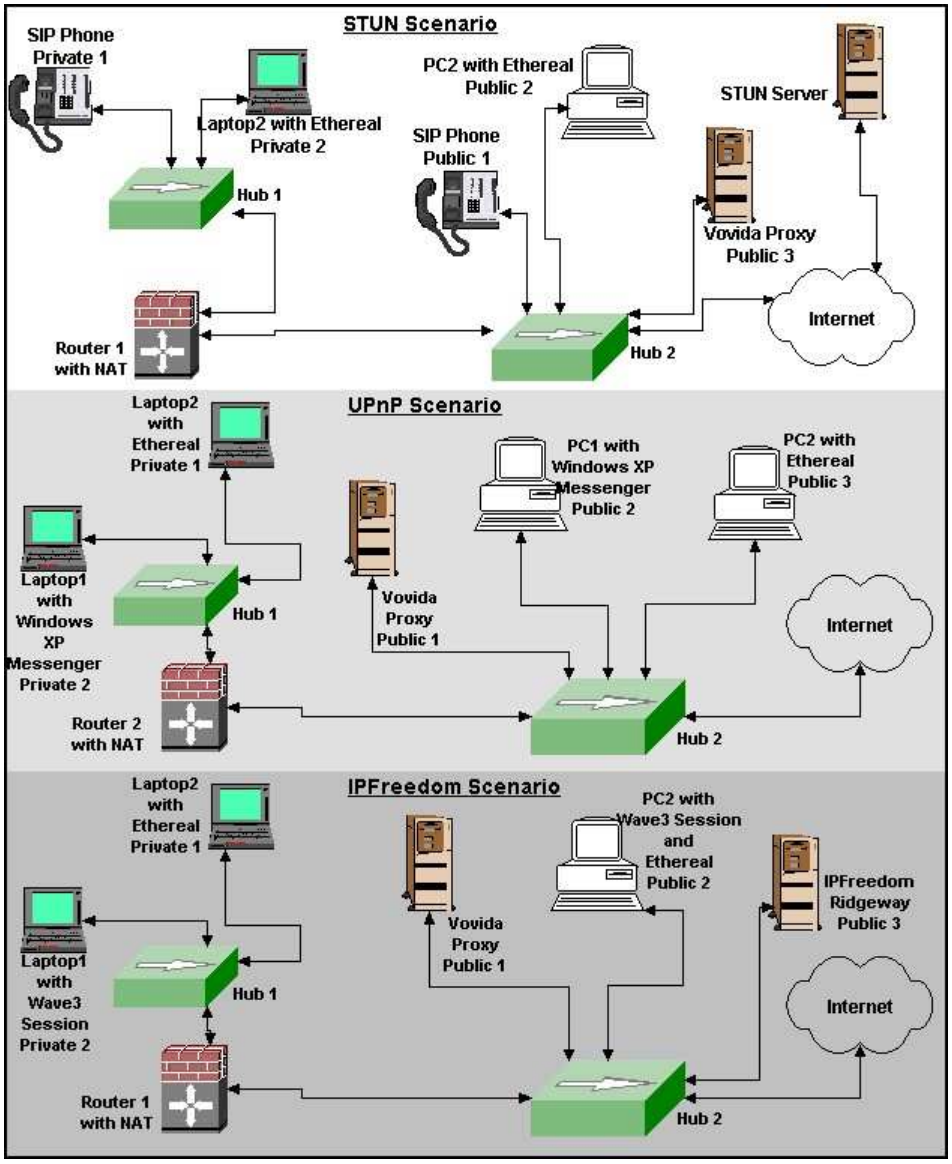

Figure 10: Small-Business Experimental Scenario Design

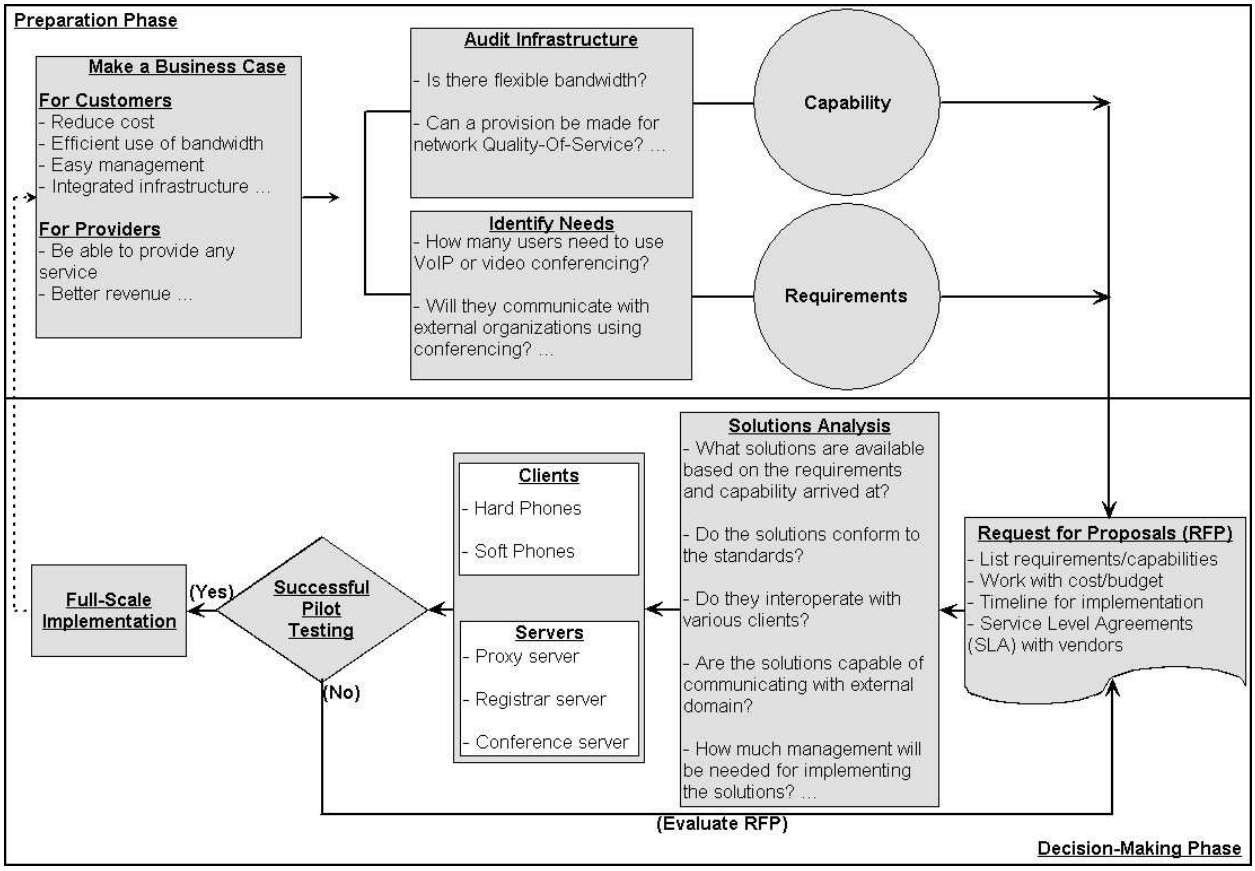

Figure 11: Management Decision Flow for SIP-based Implementations

Tables

Table 1. Performance Test Configuration

	System 1	**System 2**
CPU	Pentium4 1.8GHz	Pentium4 1.8GHz
Memory	256MB	256MB
Operating System	Windows 2000	Windows XP
Camera	Intel CS330	Logitech Express

Table 2. Performance Metrics after initiating the call

	CPU load	**Frames per second**	**Kbits per second (audio)**	**Kbits per second (video)**
System1	40-50 %	10-17	6.3/ 5.3	52.4 – 77.7
System2	40-50 %	12-25	6.3/ 5.3	65.5 -120

Table 3. Call initiation performance

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%

Table 4 – Clients Tested

Developed by	**Version**	**Referred as**
Claremont Graduate University	CGUsipClient v1.1.1	CGUsip
Cisco	Cisco IP Phone 7960 v3.06	Cisco
Grandstream	BudgeTone 100 v1.0.4.55	Grandstream
Microsoft	Windows Messenger 4.7	Messenger
WaveThree Software	Session v2.1.5	Session
Siemens	SCS-Client v1.0.0	Siemens
Columbia University	sipc v2.39	sipc
Ubiquity	Helmsman User Agent 3.0.9	Helmsman

Table 5 – Summary of Conformance Test Results

	CGUsip	**Cisco**	**Grandstream**	**Messenger**	**sipc**	**Helmsman**	**Session**	**Siemens**
SimpleTests	**3/3**	**3/3**	**3/3**	**3/3**	**3/3**	**1/3**	**3/3**	**3/3**
UAS	**4/17**	**4/17**	**6/17**	**10/17**	**6/17**	**2/17**	**4/17**	**7/17**
UAC	**17/24**	**12/24**	**12/24**	**16/24**	**10/24**	**10/24**	**14/24**	**13/24**

Table 6 – Interoperability Results

		CGUsip	Cisco	Grandstream	Messenger	sipc	Helmsman	Session	Siemens
CGUsip	**S**		Y	Y	Y	Y	Y	Y	N
	A		Y	Y	Y	N	N	N	N
	V		na	na	N	N	na	N	N
Cisco	**S**	Y		Y	Y	Y	Y	Y	Y
	A	Y		Y	Y	Y	N	Y	Y
	V	na		na	na	na	na	na	na
Grandstream	**S**	Y	Y		Y	Y	Y	Y	Y
	A	Y	Y		Y	Y	Y	N	Y
	V	na	na		na	na	na	na	na
Messenger	**S**	Y	Y	Y		Y	Y	Y	Y
	A	Y	Y	Y		Y	N	Y	Y
	V	N	na	na		Y	na	N	Y
sipc	**S**	Y	Y	Y	Y		Y	Y	Y
	A	Y	Y	Y	N		Y	Y	Y
	V	N	na	na	N		na	N	N
Helmsman	**S**	Y	Y	Y	Y	Y		Y	Y
	A	Y	N	Y	N	Y		N	Y
	V	na	na	na	na	na		na	na
Session	**S**	N	Y	Y	Y	Y	Y		Y
	A	N	Y	Y	Y	Y	N		Y
	V	N	na	na	N	N	na		N
Siemens	**S**	Y	Y	Y	Y	Y	Y	Y	
	A	Y	Y	Y	Y	Y	N	N	
	V	N	na	na	Y	N	na	N	

Table 7 – Summary of Message Counts

		STUN	UPnP	IPFreedom™
Register	Benchmark	5	9	5
	Experiment	9	85	11
Invite	Benchmark	13	13	12
	Pub>Prv	15	53	38
	Prv>Pub	15	54	34
Bye	Benchmark	4	4	4
	Pub>Prv	4	44	7
	Prv>Pub	4	44	7

Table 8 – Process Delay for each Experiment (seconds)

			STUN[3]	UPnP[4]	IPFreedom
REGISTER	Public IP Benchmark	Avg	0.03	0.43	0.11
		Stdv	0.01	0.06	0.02
	Private IP	Avg	0.07	2.78	0.11
		Stdv	0.01	0.13	0.03
INVITE	Public to Public Audio Benchmark	Avg	0.08	0.14	3.27
		Stdv	0.01	0.01	0.93
	Public to Private Audio	Avg	0.07	0.13	3.23
		Stdv	0.01	0.01	0.24
	Private to Public Audio	Avg	0.70	2.58	2.87
		Stdv	0.01	2.06	0.39
	Public to Public Video Benchmark	Avg	-	0.17	3.13
		Stdv	-	0.01	0.79
	Public to Private Video	Avg	-	0.17	2.97
		Stdv	-	0.01	0.10
	Private to Public Video	Avg	-	5.31	2.54
		Stdv	-	2.81	0.01

[3] STUN and UPnP messages are sent before the SIP messages if the client with private IP is initiating a process. Therefore, the delay measure starts from the first traversal request.

Author Biographies and Photographs

Samir Chatterjee (Samir.chatterjee@cgu.edu) is an Associate Professor in the School of Information Science and Founding Director of the Network Convergence Laboratory at Claremont Graduate University, California. Prior to that, he taught at the J Mack Robinson College of Business, Georgia State University, in Atlanta. He holds a B.E from Jadavpur University, India and an M.S and Ph.D. from the School of Computer Science, University of Central Florida. His research interests are mainly in the areas of Next-Generation Networking, Voice and Video over IP, and Network Security. Currently he is exploring fundamental challenges in designing secured IT-based systems to be used in application fields such as healthcare information systems, P2P computing, ad hoc collaboration and bioinformatics. He has published over 60 articles in respected scholarly journals and refereed conferences. He has actively contributed towards designing middleware for multimedia within Internet2 which led to the establishment of the ITU-T standard called H.350. He is principal investigator on several NSF grants and has received funding from numerous private corporations for his research. He is on the editorial board of IJBDN and JITTA. He is Vice Chair of EntNet Technical Committee for IEEE Communications Society and serves on the TPC for IEEE Globecom 2005, IEEE Healthcom 2005, IEEE MASS'05 and Workshop Chair at EntNet@Supercom 2005. He has been an entrepreneur and successfully co-founded a startup company VoiceCore Technologies Inc in 2000.

Bengisu Tulu (bengisu.tulu@cgu.edu) is currently a doctoral candidate in Management Information Systems at Claremont Graduate University. She also works as a Research Associate at the Network Convergence Laboratory. She is currently working on voice/video over IP, security, and objective/subjective quality

measurements for telemedicine applications. She has been a member of the design team that implemented CGUSIPClient, a voice/video conferencing client using SIP. She received her Masters degree in MIS from Claremont Graduate University. Earlier she received a Masters degree in Information Systems and a Bachelors degree in Mathematics from Middle East Technical University, Turkey.

Tarun Abhichandani (tarun.abhichandani@cgu.edu) is a PhD student at School of Information Science in Claremont Graduate University (CGU). His research interests include middleware for video-conferencing applications, transit-based e-government initiatives and Peer-To-Peer technologies. In the past, he has held various positions while designing and administering organization-wide networking infrastructure, database applications and ERP systems. He holds a Masters degree in Management of Information Systems from CGU and a Masters degree in Banking and Finance from Mumbai University, India. He is a Research Associate at the Network Convergence Laboratory.

Haiqing Li (liha@cgu.edu) is a doctoral student in the School of Information Science at Claremont Graduate University, and over the last three years has worked as a research assistant in Network Convergence Lab, CGU. He is also a lecturer at University of La Verne. In addition to digital signature, his research interests include network convergence, VoIP security, network simulation, and geographic information system. Mr. Li is currently working on Broadband Wireless Solutions using WiMax.