International Journal Internet Education
ISSN 1687 - 6482

http://www.IJIE.net.eg
IJIE

Case Study

DESIGN AND IMPLEMENTATION OF A SIP-BASED E-LEARNING SYSTEM

Mousa T. AL-Akhras and Iman M. ALMomani

Information Technology (KASIT), Computer Science in King Abdullah II School for Information Technology at the University of Jordan - Jordan,

ABSTRACT

Multimedia systems such as Voice over IP (VoIP) play an increasingly important role in the telecommunication world. Such systems have wide applicability in many fields and especially in areas such as e-learning or distance learning. This paper proposes and presents the design and implementation details of a VoIP system that follows the latest multimedia communication standard, Session Initiation Protocol (SIP), in its implementation. Many SIP functionalities and features for VoIP clients using recent Java technologies and APIs in the VoIP field like JAIN-SIP and the Java Media Framework (JMF) are implemented. The proposed system supports textual, audio, and visual communications between users, which allows remote interaction in distance-learning class rooms. The implementation of the system introduces an easy to use Graphical User Interface (GUI) for administrators to operate, additionally, many of the introduced features can be controlled and customised by network administrators to fit the needs of the institute that uses the system. The system is highly portable as it supports multiple platforms for all its features and services.

Key Words: Education, e-learning, java, SIP, voice over IP.

Corresponding Author: Mousa T. AL-Akhras, **Phone:** + 962 (0) 6 5355000, ext. 22602;
Fax: +962 6 5330704; **E-mail:** mousa.akhras@ju.edu.jo
Iman M. ALMomani, **E-mail:** i.momani@ju.edu.jo

INTRODUCTION

The networks of the future such as third generation and beyond networks will use Internet Protocol (IP) networks as the core transport or main carrier network. In such networks, Voice over IP (VoIP) protocol will become the main standard for third-generation wireless networks. In fact these networks are seen as the long-term carriers for all types of traffic including voice and video. This domination is envisioned by many researchers as depicted in Figure 1. Even in the existing wireless networks, IP network can be seen as an alternative[1,2,3,4].

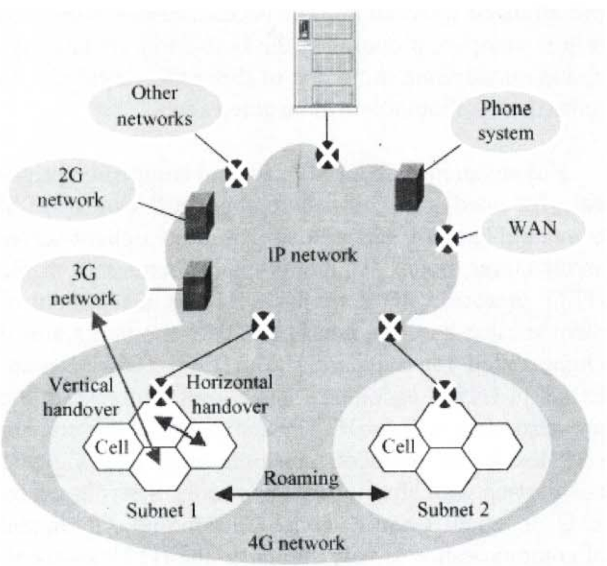

Fig. 1: Future Networks[3].

In Internet Telephony or IP Telephony technology or as it is commonly known as VoIP, a specialised signalling protocol is needed to start and terminate voice sessions and to take care of things such as accounting and charging of users as well as session management and updates. Such protocol can be used side-by-side with other Internet protocols in the TCP/IP suite such as the Transmission Control Protocol (TCP) and the Internet Protocol (IP)[5].

H.323 standardised by the International Telecommunication Union - Telecommunication standardisation Section (ITU-T) in Recommendation H.323 within the H series and Session Initiation Protocol (SIP) proposed by the Internet Engineering Task Force (IETF) in their Request for Comment (RFC) number 3261 are such signaling Protocols[5,6,7]. These signalling protocols are used side-by-side with TCP/IP protocols.

Although H.323 was standardised before SIP and has a wider share in the market, nevertheless it is a complex protocol with hundreds of headers and elements, while SIP is simpler with 37 headers which is considerably simpler and is considered to be the protocol of the future as it has adopted in Universal Mobile Telecommunications System (UMTS). In this paper, a VoIP system is proposed and implemented.

The proposed VoIP system conforms to the SIP protocol for signalling during the design, development and implementation to initiate, terminate and manage calls and sessions between two or more parties. Java technology is used to show the power of SIP protocol in VoIP technologies and to introduce many of its features that are available for use by VoIP technology. The proposed system enables users to communicate textually using the SIP standard, also it enables audio and video communications between different call parties, the proposed VoIP systems presents many features such as client authentication, user mobility and user's choice of the preferred communication quality according to the present connection speed.

The deployed system can be used as an advanced, powerful tool for exchanging textual, audio and visual content and as a remote conferencing tool used in distant-learning and remote classes applications.

Section II introduces SIP protocol and gives an overview of its architecture and session establishment sequence through the exchange of well-defined request and response messages as defined by the standard[8]. Section III reviews the features of some previously-built VoIP systems and the advantages and distinction in features of the proposed system in comparison with previous systems. Section IV gives the design details of the proposed VoIP system in both the client and the server sides. Section V discusses the implementation details of the system and the development of all of its design aspects. Section VI concludes the paper by giving an overview of the developed system and lists some of its possible applications, whereas section VII presents some avenues, features and services to be added as future work of the proposed system.

SIP OVERVIEW

SIP as standardised by the IETF in their Request for Comment (RFC) number 3261[6] is used to initiate a session between two or more users or session participants communicating in a VoIP call. It was standardised as a replacement for the H.323 protocol that was proposed earlier by ITU-T[7]. As H.323 is a complex protocols with hundreds of headers, the need arose for more flexible, simpler, and easier to implement protocol than H.323 and consequently SIP was standardised by IETF for multimedia signaling. SIP contains only 37 headers which is considerably simpler and easier to deal with than H.323 hundreds of headers. SIP does not care about the type of media to be exchanged or the type of transport. By doing this, SIP provides more flexibility than other protocols. This flexibility can be exploited to enable customised services and features. Also, SIP messages can contain optional fields that can carry user-specific information; these fields enable the creation of many intelligent and customised features.

SIP is standardized as part of the IETF's multimedia data and control architecture; therefore it can be used with other IETF's protocols such as Session Announcement Protocol (SAP), Real Time Streaming Protocol (RTSP), Session Description Protocol (SDP), and Resource Reservation Protocol (RSVP). When SIP is used in conjunction with SDP, SIP handles the communication between session participants while it relies on SDP for exchanging media capabilities.

SAP is used for advertising multimedia sessions and conferences by multicasting (Sending to many users) the session description (Defined by SDP) to a multicast address and port (Default port number is 9875). The announcement has with the same scope as the session it is announcing[8,9].

RTSP allows clients to have control over media servers by instructing commands to record and playback multimedia sessions including functions such as seek, fast forward, rewind, and pause. A user can use SIP to invite media server to a multimedia session, and then use RTSP to control operation during the session[8].

There is no point of trying to start a multimedia session, if there are no enough resources to handle this session. One of the most important resources that should be checked before starting a call is the availability of the needed bandwidth. After the call is connected it is not accepted to have variation in the voice quality, therefore, we need to check the availability of resources during call establishment. RSVP is used to reserve the needed resources.

Resource reservation can take two forms either per-session basis or on aggregate basis. In per-session basis end-to-end network resources are reserved in real-time as part of session establishment. On an aggregate basis a certain amount of network resources is reserved in advance for certain types of traffic. We need to check if the resources needed for the session to be established will not cause the total usage of that traffic type to exceed the pre-arranged reservation. The issue of resource reservation is complex proposal in the best-effort IP networks and is outside side the scope of this paper, therefore no more details about this will be given here.

Having an idea about SIP protocol companion protocols, the need is to know more about SIP standard. SIP is an application-layer protocol based on a client-server architectural model, it inherits many features from the HTTP protocol. SIP is similar to HTTP in that both of them are text-based protocols based on the International Organization for Standardization (ISO) 10646 character set in UTF-8 encoding. This allows the reuse of the programs designed for HTTP such as HTTP parsers, but such text-based protocols consume more bandwidth in comparison with binary-based protocols. SIP is based on a set of requests and responses, accordingly, its model of communication is very similar to the HTTP Protocol. Due to all of the above, SIP is seen as the protocol of the future for third generation and beyond networks due to it adoption in UMTS networks.

The client-server communication model for the SIP protocol contains a User Agent Client (UAC) that generates requests and a server that generates responses for the incoming requests. UAC is similar to the user who initiates a call, i.e. the caller. Four types of servers are defined according to the standard:

1. ***Registrar server:*** Registrar server enables UAC to register themselves. Also, users may register with a registrar server from one place, and then register from a second place. Therefore, the calls to that user can be routed to wherever the user is, which allows user's mobility.
2. ***Proxy server:*** Proxy server accepts requests from the clients or UACs and forwards them onward perhaps with some translation. For the far side, it appears as though the message is coming from the proxy rather than the original caller behind the proxy.
3. ***Redirect server:*** Redirect server responds to a request with an alternative address to which the request should be directed.
4. ***User Agent Server (UAS):*** User Agent Server accepts SIP requests and contacts the user. The response from the user is passed back on behalf of the user to the originating client (UAC). A device that implements both a UAC and a UAS can be used as a phone as it can send and receive messages at the same time. Example of such devices is SIP phones currently available in the market and used by many establishments such as multi-branches banks.

SIP includes a basic set of request messages called Methods, each method has its own functionalities. SIP requests can be sent through either the reliable TCP protocol or through the unreliable UDP protocol.

Six different methods have been defined according to the latest SIP revision issued by the IETF while other methods are also defined as extensions. The six basic methods are:

1. INVITE method used to establish a call session between users, it contains information about the calling and called parties as well as the type of media to be exchanged.
2. ACK method acknowledges the success of the INVITE request, it is used as a confirmation that the final response has been received. Just the provisional responses are not considered final.
3. BYE method ends (Terminates) the active session between the users of the call, i.e. the call parties, it can be issued by either the calling or the called user.
4. OPTIONS method used to query a server as to its capabilities. The response may indicate such capabilities as the supported media types or the availability of the user.
5. CANCEL method terminates a pending request during the initiation and establishment of the session. CANCEL can be used if the final response has not been received. If the final response has been received, the BYE message should be used instead.
6. REGISTER method identifies the location of contact of the user, it is used by the UAC to log on and register its address with a registrar server. There are many other method extensions to the SIP protocol that have been added in new RFCs such as[10,11].
7. INFO method used to transfer information in the middle of a call. Examples of such information includes:-

- Transfer of Dual-Tone Multifrequency (DTMF) digits.
- Transfer of account balance information.
- Transfer of information generated in the middle of a call from other communication networks such as PSTN.

8. SUBSCRBIE: Enables a user to subscribe to certain events.
9. NOTIFY: Used in cooperation with subscribe request and is sent when the event subscribed with happened to indicate the subscribe method has occurred.
10. MESSAGE: Used for sending instant messages rather than establishing a session that may include just sending one or few messages.
11. REFER: Enables the sender of the request to instruct the receiver to contact a third party may be to have more information or to have more accurate details regarding a subject the receiver is interested in.
12. UPDATE: If something about the session has been changed such as media type to be used, then the caller user need to inform the callee user of that change. If that change occurs after the dialog has really begun, then we can send another INVITE request. If the change occurs before we receive the final response on the first INVITE method, then sending another INVITE method will not work as it will change the state of the session. The solution is the use of the UPDATE method.

As for responses, Different classes of responses exist and each class contains one or more possible response(s). SIP responses could be provisional, success or failure, if the response is failure, the reason of the failure is indicated as well. Currently there are 6 classes that are defined in the latest SIP revision:-

1. 1xx provisional responses, provisional responses are sent as interim responses before sending the final response. This type of responses is unreliable but a new extension to the latest SIP revision suggested a reliable provisional response.
2. 2xx success responses: Currently just one message is defined in this class which is OK.
3. 3xx redirection or forwarding responses.
4. 4xx request or client failure responses.
5. 5xx server failure responses.
6. 6xx global failure responses.

All of the responses' classes can be considered final and should be acknowledged except the provisional class. For a complete list of SIP response messages one can refer to it[5].

Each response is used with its corresponding request method and has its own identification number that represents the status of the service, for example a 100 TRYING provisional response means that the proxy server is forwarding the INVITE request to the destination and it waits for a response and a 200 OK success response message indicates that a successful call session is established. Figure 2 shows a flow diagram of a simple SIP messages' exchange occur during an establishment of a call session.

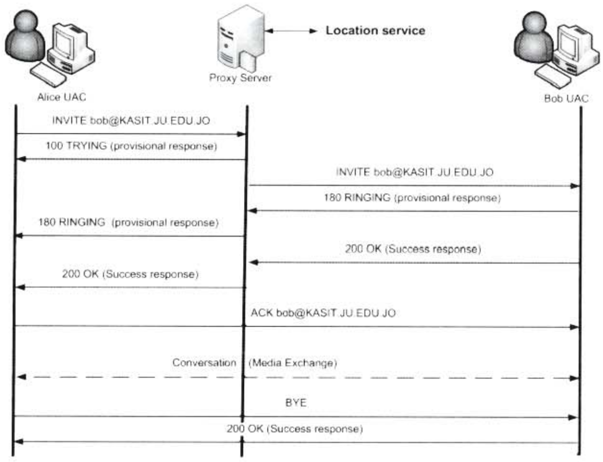

Fig. 2: An example of message exchange in SIP.

In the above diagram, Alice invites bob to a new session which could be a phone call or a remote lecture, the invitation is implemented through the invocation of the SIP's INVITE request. The request is sent from Alice's UAC to the proxy server, the proxy server send a TRYING interim response to Alice's UAC to indicate it starts the process of forwarding the invitation to bob. The proxy server first solicits a location (Registrar) server about bob's current location, once updated of his current location, it then forwards the message to bob. Bob's UAC may respond by sending one or more provisional responses such as RINGING interim response, to indicate that bob's is being alerted, to the proxy server which in turn forwards the message to Alice's UAC.

Once Bob's accepted the invitation by either picking up the phone or may be by pressing a button or another graphical element to indicate his acceptance of the invitation, an OK response is sent from bob's UAC to Alice's UAC through the proxy server. Alice's UAC then send an acknowledgment message (ACK) to Bob's UAC. When received by Bob's UAC, the conversation or media exchange which could be textual, audio, or visual content may begin.

At the end of the call one of the call-participants hangs up, which causes a BYE message to be sent from that user's UAC to the other user UAC. The party receiving the BYE message sends OK to confirm receipt of the message and the session is over.

SIP addressing is based on the use of Uniform Resource Indicators (URIs) to address entities in requests and responses. URIs are very similar to the usual Uniform Resource Locators (URLs) used in the internet. The URI address can take the form user@host. When SIP is to interwork with telephony network PSTN, the user portion of the SIP address could be a telephone number which greatly simplify the task of mapping telephone numbers to URIs. In this schema the address could look like sip: 0096265355000@KASIT.JU.EDU.JO.

Number of headers are included in the SIP's message's header. These headers provide additional information about the message. Some headers are used just in requests and others just in responses. Some of message's Headers are mandatory and some are optional. Among the mandatory headers are: The message originator and the intended recipient.

A complete list of SIP message's headers and a detailed overview of the SIP protocol architecture and a detailed technical and usage comparisons with the H.323 can be found in the literature[5,12,13,14].

RELATED WORK

In the past few years many VoIP systems have been built based on SIP standard, H.323 standard or even based on other proprietary protocol such as Skype[2115]. Quite successful, commercial systems were built using such protocols and complete companies such as Skype are employing hundreds of employees to develop and market such systems.

Previous systems use some functionalities of their proprietary protocols or some other functionalities according to the authors of the papers whereas the proposed system in this paper conforms to the SIP protocol in collaboration with its companion protocols such as Session Description Protocol (SDP) and Real-time Transport Protocol (RTP) to ensure that the system follows the standard and make it compatible with other standardised systems.

Many features are deployed in our system making it distinguishable from other work: Most notably is the ability to communicate using either text, audio or even video conference. The users are able to use different media formats to choose the best codec that fits their needs according to the network conditions as will appear in section IV. The ability to communicate via video conferencing could be used as a remote conferencing tool in distance-learning system which something did not appear in previous implementations. Also, many extra functionalities and features are implemented at the server-side to enable advance administration such as conference management, network monitoring, client authentication, database management, registration server, proxy server, and location server. Many of the above features are not available in previous implementations.

Some of the previous implementation were built using Java update of the JDK 1.2 where the functionalities are limited to the non-reliable UDP communication and the VoIP implementation did not fully follow the SIP standard as it was released in its current form[16]. In this paper the latest version of JDK 1.6 is used for implementation and the user is given the option to use either the reliable TCP protocol or the unreliable UDP protocol and SIP protocol is fully followed in the current implementation.

In other applications many of the important features are missing such as video conferencing, the possibility of encoding voice packets according to the user's selection based on network congestion conditions. Rather than giving the user the facility to select the best encoder according the available bandwidth, voice packet were not encoded at all or if it is encoded, it was encoded using one possible encoding algorithm according to the implementer's choice. Additionally, no session timer for audio and video calls is used[17]. Java Media Framework (JMF) is also used in the current system. Some of the previous implementations and their main features are reviewed next.

Ohshima et al.[18] proposed doing media mixing of incoming media streams, in case of conferences, at the server side then the mixed media is sent back through the network to the users. A different approach is followed in the current implementation as the mixing is based at the receiver side where each user send his/her stream and the mixing occurs at the receiver-side. This is due to the fact that each user may have different codecs for the media streams.

The system proposed by Ohshima et al. may increase the bandwidth allocation of the network because of the heavy load of mixed media sent from the users to the server and then sending the mixed stream back to the clients while as our system is based on client-server architecture, the media streams are mixed at the user side which decreases the load on the network. Additionally, Ohshima's approach suffer from the problem of single point of failure while in our system if one machine fails, the other can proceed with their sessions.

More details about previous systems are given later in this paper appropriately when a comparable approach is discussed in the current implementation.

SYSTEM DESIGN

The system is designed according to the SIP standard of a client-server architecture[6], together with the support and added features of other protocols such as SDP[19], RTP[20] and JMF that was used for the multimedia features, a fully SIP-Supported client-server VoIP system is implemented that provides many features for VoIP clients to help improve their multimedia communications and call management experience. The system design approach has two main compartments: client design and server design.

A. Client Design:

The client application adheres to the SIP protocol and is built according to the standard. The client application provides a Graphical User Interface (GUI) to enable its users to operate and control the application's features conveniently, also users have the ability to easily modify the SIP settings of their application according to their SIP provider's connection settings.

Users of the client's application are able to make either peer-2-peer calls or conference calls using audio, visual or textual communications and they have a choice of different supporting codecs for audio and video communication. The choice of the codecs affect the quality of the session and it depends on their connection speed and the available bandwidth for their application.

In addition, users of the application have the ability to save their personal information, address books and SIP settings in their personal profiles securely using a password authentication process to verify their identity along with their profile name. As it will be detailed in the section V, the client side design is divided into 4 components:

1. ***Graphical User Interface (GUI):*** The friendly interface that enables different users to interact easily with different features available at the client side.
2. ***SIP Controller:*** Handles SIP messages, responsible for sending and receiving SIP messages.
3. ***Media Transceiver:*** Responsible for the sending and receiving audio and video contents.
4. ***User Profile and Authentication:*** This component controls saving and retrieving user's information and providing security for user's personal information through password authentication.

Different components that comprise the client are shown in Figure 3.

B. Server Des.ign:

During the server design, SIP standard was carefully considered. The SIP Protocol defines three types of servers: Proxy, Registrar and Redirect servers (Used in case more than one domain co-exist) in addition to the UAS. As explained earlier, the proxy server is responsible for forwarding SIP messages to the clients in its domain, Registrar server is responsible for registering the clients with their real location of contact during the registration process, redirect server helps in redirecting SIP messages to destinations outside their corresponding SIP proxy domain, in addition to the above servers, the current server design includes extra features and services to provide administration control over the service and to provide better efficiency and stability of the provided services.

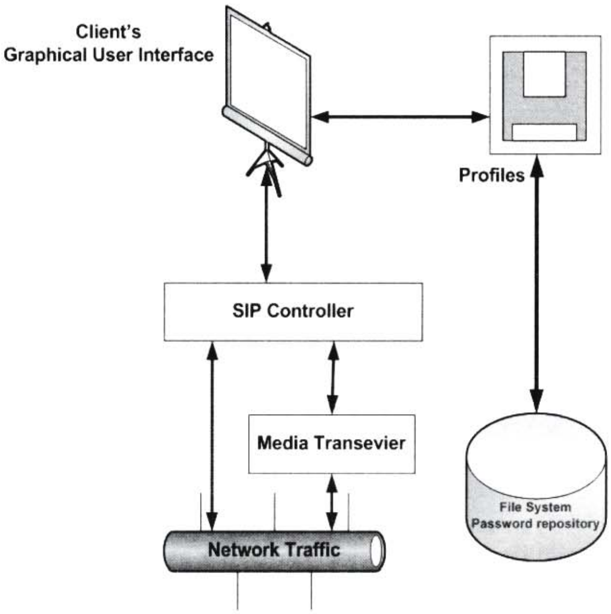

Fig. 3: Layers of a VoIP SIP Client.

The designed SIP server supports user's mobility in its administered domain to enable free movement of users with no restriction on their location or connection mode (Wired, wireless, LAN, WAN, etc.) within the server's domain, the server also provides the security of user's identities through password authentication of users where they have to provide login information of their SIP accounts using their acquired SIP's URI address and password for authentication during the server registration process of users, it is also responsible for monitoring conferences taking place in its administered domain, where it keeps a list of these conferences for the system and network administrators, it keeps track of the conference information and current users in all the conferences and the initiators of these conferences. Different components that comprise the server are shown in Figure 4.

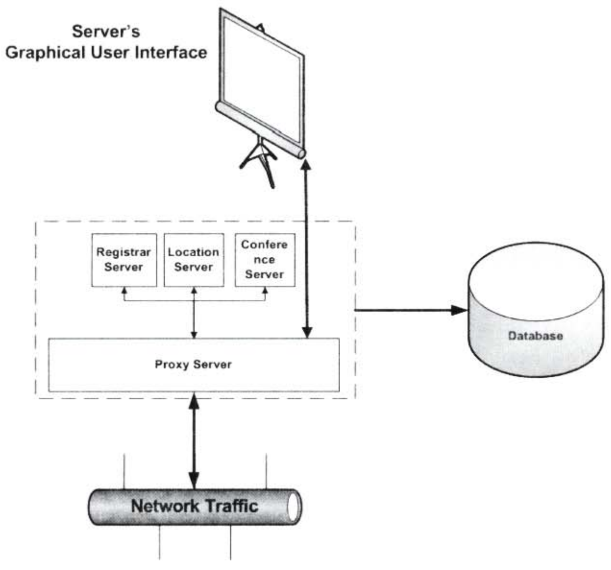

Fig. 4: Layers of VoIP SIP Server.

The design architecture of the server includes the following elements:

1. Graphical User Interface (GUI): The friendly interface that enables network administrators to easily interact with different services available at the server side.
2. SIP Controller: Handles SIP messages, responsible for sending and receiving SIP messages.
3. Database Manager: Responsible for database connectivity, accessing database entries as required by different services. Developed using Derby Database Management System (DBMS).
4. Conference Manager: Responsible for monitoring and managing ongoing conferences.
5. Registrar Server: Responsible for registering users at the proxy server and accessing online status of different users.
6. Proxy Server: Responsible for forwarding messages to corresponding users in the current domain.
7. Location Service: Responsible for recording the location of users during the registration process.
8. Network Monitor: Responsible for monitoring SIP traffic over the network.
9. Client Authentication: Responsible for checking the client's registration details to make sure of the clients-claimed identity.
10. Availability: Checking the on-line status of users before forwarding the messages.
11. Mobility: Forward the message to current location of the user.

SYSTEM IMPLEMENTATION

The client-server VoIP system was developed and implemented using Java programming language because of its efficient application development, object-oriented concepts and component modularity features, which supports the division of the application components and improves the intended application features. Also as Java is platform-independent, therefore it provides the portability of our system across different platforms that can be found in today's networks. This removes any restriction on the user's operating system or used browser and reliefs us as developers of the system from developing different versions of the system to different user types.

Many of the non-intrinsic Java features are also available through its basic or third-party APIs. The Java community has played a major role in the changing of the telecommunications market from large, proprietary, closed systems to an open architecture of rapidly deployable services, based on this initiation a Java Application Programming Interface (API) called Java API for Integrated Networks (JAIN)[21] is developed and called the JAIN initiative[22]. JAVA has a defined a set of Java technology APIs that enable rapid development of Java-based next generation telecommunications products and services especially in the telecommunication markets and industries. The Java APIs defined through the JAIN initiative bring service portability, network independence, and open development to telephony, data and wireless communications networks. One of these APIs which is used for our development is the JAIN-SIP API. JAIN-SIP has been developed by the National Institution of Standards and Technology (NIST) as a Protocol Stack implementation for the SIP protocol developed purely using the Java language to be used as a Reference Implementation for Java developers who require Java-based SIP APIs for their SIP-based developed systems and applications using Java. The architecture of JAIN-SIP is based on a consumer/producer model, where a SIP message is produced by the sender and consumed by the receiver1. JAIN-SIP architecture is shown in Figure 5.

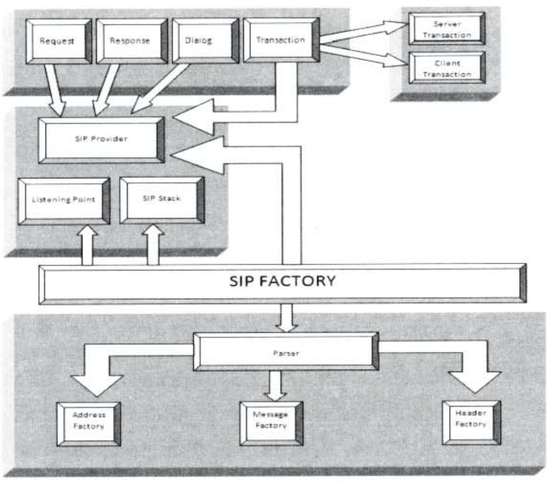

Fig. 5: JAIN-SIP Architecture.

The JAIN-SIP API structure is mainly based on Factories, the main factory which is the SIP Factory is the base of the SIP implementation, this is where all the SIP settings are set and a reference implementation for the SIP protocol is used, the sub-factories that are generated from the SIP Factory are the Address, Message and Header factories, together they help in building and parsing SIP messages that are delivered and sent throughout the network.

To send and receive SIP messages across the network, a listening point is needed, this is generated from the SIP factory, to handle SIP messages, a SIP stack is needed, the SIP stack hooks-up the listening point with the SIP Provider to handle the SIP messages and perform operations according to the protocol standards and its implementation in the API, the SIP Provider handles Request and Response messages, and is responsible for the creation of Dialogs and Transactions of created sessions.

Alongside with JAIN-SIP API, SDP protocol is also used to describe the SIP's session preferences, multimedia capabilities and communication information that is used during call setups using SIP's INVITE method in peer-2-peer calls and conferences, an API reference implementation is also included with the JAIN-SIP API package for SDP protocol. Real-Time Transport Protocol (RTP) is used for multimedia transmissions under the Java Media Framework (JMF) which also provides great support for media packetizing, multiplexing, demultiplexing and other multimedia control and session management features as well as many different supported codecs provided by the framework for audio and video multimedia communications.

The client-server VoIP system was implemented using the Java Development Kit (JDK) version 6, update3. NetBeans IDE version 6 is used for development, the developed system has been tested and is fully operational on Microsoft Windows XP as well as Microsoft Windows Vista with Pentium 4 Processors and 512 MB RAM using the Java Runtime Environment (JRE) version 6, update3. The client application size is less than 2 MB and the server application is less than 500 KB, the Derby DBMS running at the server side is less than 30 MB, therefore the system has high memory efficiency and supports platform-independent system installation due to the use of Java as the development language.

Next both client implementation and server implementation are given in details.

A. Client Implementation

Recall that in section IV, the set of components that made up the client application in the system were defined as following:

1. Graphical User Interface (GUI).
2. SIP Controller.
3. Media Transceiver.
4. User Profile and Authentication.

The following describes the implementation process of each of the above components:

A) Graphical User Interface:

The GUI design required the use of the Java Swing Framework which has been included as a standard API package distributed with JDK version 1.6 and newer, the GUI includes three windows:

- Main Window.
- Peer-2-Peer Call Window.
- Conference Call Window.

Each of these windows is explained in more details next:

- Main Window: The main window includes two buttons, Load User Profile and Sign In. Load user profile loads the user's profile, this includes an authentication process where the user input his/her profile name and password for that profile, profiles are stored as files where each user has a folder that includes the SIP settings, personal information and address book as files. If the user's profile does not exist, a new profile is created and added to the set of stored profiles for users, a menu bar is included in the main window GUI to provide an ease of use of the main window's functionalities, once the profile is loaded, the user can change his/her profile settings and other settings easily through a drop down menu from the menu's tool bar that opens the settings sub-window.

The main window consists of 3 tabs: account information tab, call tab, and Address Book tab. The 3 tabs are used to display SIP account information once connected to the SIP Proxy, for making peer-2-peer calls and initiating/joining conferences, and for adding, deleting and modifying entries in the address book of the user, respectively. Figure 6 shows the client Interface in 3 parts to show different tabs.

Fig. 6: Client Interface: a) Account Information. b) Call tab. C)Address Book tab.

- Peer-2-Peer Call Window: When the user decides to make a Peer-2-Peer call to a certain SIP user, a new window is created for that call, users can make multiple Peer-2-Peer calls simultaneously without the worry of any interference between their different call windows and SIP sessions, the window contains a menu bar for different options and call information requested by the user about the session, users can end the call either by the corresponding option from the menu bar or by pressing the Hang Up button.

The window includes a text area where the call status and different SIP session setup messages are displayed for informative use, another text area is used to enable users to communicate textually with each other through text messages, the window includes Audio start and stop buttons for enabling and disabling Audio communications and Video start and stop buttons for enabling and disabling Video communications, also this window gives users choice of different themes to choose from according to their style and mood. An Example of peer-to-peer call is illustrated in Figure 7.

Peer-to-peer calls can be used for making one to one calls or one-to-one personal lessons. In the e-learning context, students can post their question either in textual format or in a form of verbal question. The lecturer can make demos and even board explanation making use of available camera transmission, this requires no extra hardware at the student or receiver side.

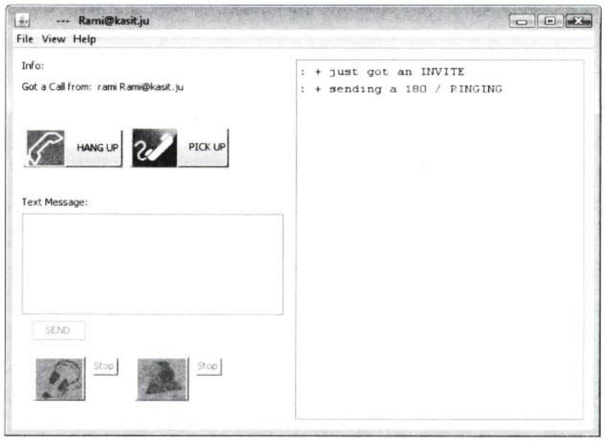

Fig. 7: An example of Peer-2-Peer Call.

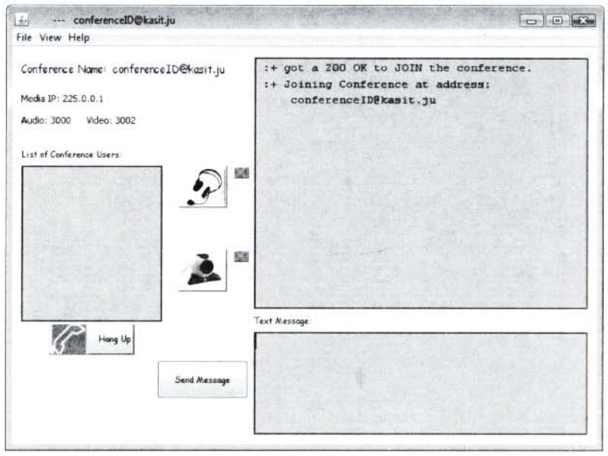

Fig. 8: An example of Conference Call.

- Conference Call Window: Once a user has joined a conference, a new window for the conference call appears, users can join only one conference at a time, multiple conference windows is not supported due to the logical concept that users can not be in two conferences or class sessions at the same time.

The conference window includes an information window that displays information about the conference, session information, media communications, list of users in the conference (Conferees) and the conference name, also a text area is included in the window for the same purpose as in the Peer-2-Peer call window, Audio and Video start and stop buttons are also included in the conference window to enable and disable their corresponding communication methods, this window also provides different themes to users to choose from according to their different tastes and styles. An Example of conference call is illustrated in Figure 8.

In the e-learning context, students can subscribe to an event which could be something like the start of a lesson. When the lesson starts, the server may notify the subscribed users of this event. In the conference window multiple students can listen to one lecturer. Even two lecturers not necessarily at the same place can participate in explaining the same lesson.

The second component in the client's implementation is the SIP controller. This is the backbone component for both the client and the server side applications, this controller is responsible for creating and handling of the exchanged SIP messages and keeping track of all the SIP sessions and status for ongoing sessions. SIP controller is responsible for creating, initiating, modifying and tearing down SIP sessions and connections, this controller uses the JAIN-SIP API package for its operations and implementations.

The SIP controller settings can be modified through the settings window from the main window, the settings that are available for the users are the port number of which the SIP protocol will operate on, and the transport protocol that can be used with the application, the provided choices are UDP and TCP transport protocols, both protocols are fully supported in our client-server VoIP system, settings for connection to the SIP Proxy are also available in the same window.

To begin with the operation of the SIP controller, the Sign Up button from the main window will open up a login window for the users to provide their SIP URI and password for the authentication process at the server side, this process is the registration process which is sent as a REGISTER (SIP standard) message that includes all the information needed for the registration process at the server side, if the process completed successfully a 200 OK response message will be received at the user's SIP controller, otherwise a corresponding response message will be received which is then displays to the user the error that occurred during the registration process. After the registration, if successful, the user will have the ability to begin making VoIP calls.

To initiate a peer-2-peer call, an INVITE message is created with user's information along with the destination's SIP URI, also the SDP information is included with the content of the INVITE message as a 'SDP/application' MIME type as its content type, the SDP includes the session and media information needed to create the VoIP call successfully, if the INVITE message is received at the destination, a 180 RINGING provisional response is received, once the SDP negotiation for session settings (Codec type, multimedia content, etc..) is completed and the destination has answered the call, a 200 OK response is received, if the destination hangs up the call before picking up, a BUSY response is received, if the user hangs up, a CANCEL request is sent to the destination and the call is terminated, if the call was ended by either sides after it is answered, a BYE request is sent and a 200 OK for the request will be received.

As for conference calls, we made use of SIP's Protocol extension Session Timer for users to join and leave conferences, a SIP INVITE request message is created by any user who want to join the conference with the corresponding SDP information included in its content, the message header Expire-Header field is used as a reference for the SIP Proxy to determine whether the user is joining or leaving the conference.

The SIP controller has a context controller to keep track of all the SIP sessions that are initiated and in progress and all of their corresponding call windows created for their sessions, the context controller is responsible for delivering the control SIP messages to their corresponding calling windows created for their session, this controller helps in the creation of the multiple peer-2-peer call windows and to prevent the interference of the control messages that passed between these windows and their sessions.

3) Media Transceiver

The media transceiver is responsible for the multimedia communication between the users in peer-2-peer and conference calls, it includes the Audio and Video communications, this component is implemented through the use of the Java Media Framework, this framework includes the multimedia processing and session management, and also packetizing multimedia communication contents sent across computer networks through the use of RTP and RTCP Protocols.

The settings that are available for the users in the settings window include the Audio and Video ports used for multimedia communications using the RTP Protocol, and also the codec to be used for Audio and Video communications, the user has a choice of the following list of codecs that are supported by the Java Media Framework (JMF):

Audio: G.711, GSM, G.723 and DVI.
Video: JPEG, H.261 and H.263.

The media transceiver is responsible for delivering and receiving audio and video communications to and from the clients of the session, peer-2-peer calls handle the multimedia communications in a direct (Unicast) connection mode. In conference calls, the multimedia communication is handled in a multicast connection mode, this provides better network efficiency and less traffic load between the users and optimises the connection bandwidth of the conferees, other solutions were proposed[23-27] that are based more on a server-side approach, the proposed solutions require client routings, path calculations, distribution management of connections, these require highly sophisticated development measures at the client side and require higher bandwidth for control and management messages that are sent over the network alongside with the multimedia traffic, the above solutions did not use the processing potentials available at clients for media processing. In our proposed system, the multimedia conference approach is based on multiplexing and processing of the multimedia content and connections at the client side instead of the server-based approach, standard today's computers have dual- or quad processors with high memory capacity which is similar or may be exceeds the power available for supercomputers few years ago, therefore today's users will not need to worry about the available processing power needed for the multimedia processing as was the case in old technologies, therefore using the multicast solution combined with the per-client multimedia processing is the most efficient, bandwidth economic and network optimised solution for multimedia conferences taking into consideration current technologies.

4) User Profile and Authentication

The client application provides users the ability to save their personal settings, along with their predefined SIP and multimedia settings and their address book entries to provide them the ease of use of the application, they are all stored in files on their system for later use, the information is stored and protected using a password authentication process where users will have to provide their profile name and password to access the profile information when loading their profile from the application window, in case the profile does not exist, a new profile is created and stored in the system, the storage structure is folder based, where the profile name is used as the user's folder name and the information of the profile is stored as files.

B. Server Implementation

The server application includes a SIP proxy server in addition to the Derby DBMS for extending the server's features and supporting its operations and provided services, the system components that were identified in the design stage are implemented and developed, the following sections describe the implementation and development of each component defined during the design stage.

1) Graphical User Interface

The GUI of the server application introduces a tabbed window to show all available services and features supported by the SIP server, there are 4 tabs:

- ***Traffic Monitor tab:*** This is where the server administrator can keep track of all the SIP messages that are sent and received by the SIP server during its active online session as it is displayed in an informative text area.

- ***Location Service tab:*** Displays all users that registered with the SIP Server and are currently online, alongside with the information of each user, their display names, SIP URIs and the current IP address that they are registered with as their real location of contact.

- ***Conference manager tab:*** The place where the conference monitor information is displayed, it displays all the information of currently active conferences, and each conference has its own informative display window to display the current users in a specified conference.

- ***Settings tab:*** The place where the SIP server settings can be defined by the network administrators, this window is also used to activate the SIP server and restart it in case of any error or malfunction.

2) SIP Controller

As in the client application, the server application also has a SIP Controller as it is the backbone of all the operations and services of the system.

The SIP methods that are supported by the server:

- INVITE
- CANCEL
- BYE
- ACK
- MESSAGE

The SIP responses that are supported by the server:

- 1xx Provisional Responses: 100 Trying, 180 Ringing.
- 2xx Success Responses: 200 OK.
- 4xx Client (Request) Errors.
- 5xx Server Errors.
- 6xx Global Failures.

Other requests and responses can be added in future versions of the current system. The SIP controller also includes a context controller for management of multiple sessions, this controller helps in preventing the interference of a certain session's control messages with other sessions, this helps the server in providing better performance and stability as the number of users and sessions increases.

3) Database Manager

This component is responsible for supporting the main services of the SIP server, this includes the storage of all the SIP clients of the current SIP server's domain and their information including their online status, registered IPs, account information, information for client authentications, conference listings and temporary information storage for these listings and conference monitoring.

The database used to store information is Apache's Derby DBMS which is a very light weight and highly portable DBMS, it is fully supported by Java through the use of the Java Database Connectivity (JDBC) driver, this database provides very powerful features and tools that help in the productivity and scalability of the SIP server as well as supporting of all its services and features and provides the performance and stability of its operations.

Alongside the DBMS, a customized database viewer and editor has been developed for the server administration purposes to make any necessary modifications to the database system without the hassle and complexity of the extra information needed to make changes in current database GUIs such as adding or modifying entries, the customized viewer and editor provide an easy to user interface and many useful functionalities for system administrators to make any required changes.

The database is used to support the location service, conference management, client registration and authentication features of the SIP server.

4) Conference Manager

This component is responsible for monitoring and management the ongoing conferences and keeping track of each of the conferences information and settings, it is also responsible for the creation, setting up and deletion of conferences and their sessions.

When a user sends an INVITE request to join a conference, the conference manager component takes the conference information from the SIP request message and checks whether the conference exists, if it exists, then the session and communication information and details are sent back to the user with the list of users that are currently in the conference and the conference manager adds the user to the users' list of the conference that is stored in the database. On the other hand if the conference does not exist, then the conference manager creates and sets up a new conference session, and informs the user of the new session's information and communication details, the conference manager creates a new conference entry in the conferences' list with the conference's name and settings and the creating user as its initiator, the manager will also create a new storage table for the list of users of the newly created conference.

When a user sends a SIP request message to leave a conference, the conference manager removes the user's entry from the user list of the corresponding conference that are mentioned in the SIP request message, when the last user leaves the conference, the conference manager removes the storage table of the conference and removes the conference entry from the conference list as it is no longer an active conference.

Updates to the list of users is sent to all users of a conference whenever the user list of the corresponding conference has been modified either by joining or leaving users.

The settings of this component that are available for the server administrators are located in the settings tab of the server GUI application, the options available are the multicast settings of the conferences, the multicast IP used for multimedia communications and the number of conferences that are limited by the range of ports provided by the network's administrators.

5) Registrar Server

The registration server is responsible for registering clients of the SIP server's domain, clients send REGISTER requests to the SIP server which is then sent to the registrar server, the REGISTER method includes and Expire-Header field to notify the SIP server with users that are registering or unregistering from the SIP server. This component then abstracts the client information from the SIP message request and checks its existence and consistency with the client's entry in the server's database and the expire-header field to determine the registration process.

The registrar server has an option available at the settings tab of the SIP server's GUI to enable or disable automatic client registration at the server, this option indicates whether clients that register at a certain SIP proxy's domain with no entry in that domain's database server are able to be automatically register as new clients at the server or whether their entries must be manually entered by the domain's administrator and new clients who try to register but has no database entry will be denied.

Also, the registrar server only registers clients from the domain that it administers and registration requests that belong to domains that are not administered by this server will be rejected and clients will be informed of an invalid domain registration request.

6) Proxy Server

The proxy server forwards SIP messages-whether they are requests or responses- between SIP clients in the server's domain, the messages include all the SIP messages that are supported by the SIP server. It acts as a relay point for the SIP control messages between the SIP clients in their corresponding sessions.

7) Location Service

The location service with the aid of registrar server helps in locating the real address of contact of the SIP clients who are currently registered and online. Once the user sends the REGISTER message to the SIP server, the message will be processed by the registrar server, after processing, the request is sent to the location service to abstract the contact information of the registered client, this information will later on be stored in the database system as its real address of contact, this includes the SIP's IP address and SIP port of which the client used to register itself with the SIP server and at that client's entry, a currently online status will be set. When the client unregisters from the SIP server, the online status of that client's entry will be replaced with an offline status.

8) Network Monitor

Network monitor provides the server and network administrators, a SIP message traffic monitor facility to monitor the SIP messages that are passed through the SIP's server, all SIP messages are displayed on a history pane in the server's GUI where administrators could study these SIP messages and monitor the SIP traffic over the SIP server's domain.

9) Client Authentication

Security features have been included to the SIP server to provide authentication and client confidentiality, the database is encrypted using the DBMS's encryption technique of its storage, transactions and connection mode, as for authentications, clients provide their account name and password in the REGISTER request message for the authentication process at the server side, password authentication is used to authenticate accounts stored in the server's database, once authenticated, the client is registered and considered to be online.

10) Availability

As for users availability, when clients create INVITE requests for certain user, the status of the invited user is first checked by the server to see whether the user is online and available before forwarding the message to the destination, if the user is not available, the user who issued the INVITE request will be informed of the offline status of the invited user.

11) Mobility

Users are registered at the registrar server to verify their status and register their real address of contact of which the users use to create VoIP sessions and connections, the real address of contact is automatically updated every time the user sends a REGISTER request to the SIP server, this provides the users with the mobility where users can connect from anywhere within its domain that it is administered provided with VoIP and SIP services.

CONCLUSIONS

In this paper, the design and implementation details of SIP-based services are provided. The proposed VoIP system has provided and implemented many SIP functionalities and features for VoIP clients using recent Java technologies and APIs in the VoIP field like JAIN-SIP and the Java Media Framework. The proposed system supports textual, visual and audio communications between users, it can used in scenarios like remote class room and VoIP calls. The system developed introduced the processing power of the SIP clients and their systems for the multimedia content and their possible potentials using today's technologies. Also security measures have been implemented through password authentications of SIP clients, the server operations and services are designed and built according to the administrator's preferences, they provide an easy to use interface for administrators to operate, control and customise settings. The system is highly portable and supports many multi-platforms for all its features and services.

FUTURE WORK

As for future work, more SIP methods such as UPDATE, REFER, NOTIFY are to be added and implemented in the system, also newer codecs could be added for example the G.729 for better audio compression at higher qualities, more security restrictions could be added such as policies, blacklists, call filtering of SIP users and calls could be added alongside with SIP message encryptions and better authentication process could be implemented using the latest techniques.

More value-added services such as billing of SIP users and providing a PSTN gateway for landline phone calls are possible add-ons for our current implementation of VoIP system.

REFERENCES

1. Lieve, B. And Suresh, L. 2001. Toward An All-Ip-Based UMTS system architecture. IEEE Network 15(1): 36-45.

2. Goetz, I. 2004. VoIP interconnect and quality impacts on 3G core network evolution. IEE Conference Publication (503): 534-536.

3. Heiman, F.P. 1998. Wireless LAN Voice over IP telephone system. Northcon Conference Record: 52-54.

4. Svanbro, K., Wiorek, J. And Olin, B. 2000. Voice-over-IP-over-wireless. In The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Sep 18-21, Anonymous, 24-28.

5. Daniel, C. 2003. Carrier grade Voice over IP. 2nd. McGraw-Hill.

6. 6- Session Initiation Protocol (SIP). Session Initiation Protocol. 2002, May, Internet Engineering Task Force (IETF).

7. Packet - Based Multimedia Communications Systems. International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) Recommendation H.323. 2003, Jun.

8. Nachiappan, N. and Sjoqvist, F. Survey of Voice over IP (VoIP). 2004. www.geocities.com/stanford_nachi/VoIP_Survey.doc. Anonymous.

9. Protocols.com. Voice over IP. 2008. http://www.protocols.com/pbook/voipfamily.htm. Anonymous.

10. Session Initiation Protocol (SIP). Update. 2002, Sep, RFC 3311, Internet Engineering Task Force (IETF).

11. Session Initiation Protocol (SIP). Specific event notification. 2002, Jun, Internet Engineering Task Force (IETF).

12. Glasmann, J., Kellerer, W. And Muller, H. 2001. Service development and deployment in H.323 and SIP. IEEE Symposium on Computers and Communications Proceedings: 378-385.

13. Park, S.O., Han, J.C., Lee, I.J., Kang, S.G. And Ho, L.H. 2007. Mechanism of audio-text mode assignment on SIP-based internet telephony system. International Conference on Advanced Communication Technology, ICACT 2: 1294-1297.

14. Spinsante, S., Gambi, E., Perotti, A. And Vespasiani, A. 2004. A Java-based VoIP application extending LAN telephony facilities in a MGCP framework. IEEE 6th Workshop on Multimedia Signal Processing: 454-457.

15. Skype.com. White Paper - JAIN TM and Java TM in Communications.(old 21)

16. Zou, H., Wang, H., Mao, W., Wang, B., Focant, S., Handekyn, K., Chantrin, D. And Marly, N. 2000. Prototyping SIP-based VoIP services in Java. In International Conference on Communication Technology WCC-ICCT, Anonymous, 1395-1399.

17. Zeadally, S. And Siddiqui, F. 2004. Design and implementation of a SIP-based VoIP architecture. Proceedings International Conference on Advanced Information Networking and Application (AINA) 2: 187-190.

18. Ohshima, K., Tajima, Y. And Terada, M. 2004. A teleconferencing system with high-speed stream mixing for Voice over IP. Proceedings International Symposium on Applications and the Internet: 295-298.

19. Session Description Protocol (SDP). Session description protocol. 2006, Jul, Internet Engineering Task Force (IETF).

20. Real-Time Protocol (RTP). A transport protocol for real-time applications. 2003, Jul, Internet Engineering Task Force (IETF).

21. Al Canaan, A. And Khoumsi, A. 2008. Advanced Ip-Telephony service creation using JAIN-SIP API: Cross-platform approach. In Mosharaka International Conference on Communications, Networking and Information Technology, MIC-CNIT'08, Amman, Jordan, Anonymous.

22. Sun.Com. White Paper - JAIN TM and Java TM in Communications, March 2004.old23

23. Gu, X., Nahrstedt, K., Chang, R.N. And Shae, Z.Y. 2004. An overlay based QoS-aware Voice-over-IP conferencing system. IEEE International Conference on Multimedia and Expo (ICME) 3: 2111-2114.

24. Chevillat, P.R. And Schott, W. 2003. Broadband radio LANs and the evolution of wireless beyond 3G. IBM Journal of Research and Development 47(2-3): 327-336.

25. Prasad, R.V., Hurni, R. And Jamadagni, H.S. 2003. A scalable distributed VoIP conferencing using SIP. In Eighth IEEE International Symposium on Computers and Communication (ISCC), Anonymous, 608-613.

26. Sat, B., Huang, Z. And Wah, B.W. 2007. The Design Of a multi-party VoIP conferencing system over the internet. In Ninth IEEE International Symposium on Multimedia (ISM), Anonymous, 3-10.

27. Yang, S. And Chen, W. 2008. Sip multicast-based mobile quality-of-service support over heterogeneous IP multimedia subsystems. In IEEE Transactions on Mobile Computing, Nov, Anonymous, 1297-1310.