Interactive Distance Learning based on SIP

S. Sae-Wong, T. Kamolphiwong, S. Kamolphiwong, and N. Wittayasirikul

Centre for Network Research (CNR),
Department of Computer Engineering,
Faculty of Engineering,
Prince of Songkla University, Hatyai, Thailand 90112
suthon@coe.psu.ac.th, kthossaporn@coe.psu.ac.th , ksinchai@coe.psu.ac.th,

Abstract

In this paper, we present a simple design framework of IDL (Interactive Distance Learning) based on SIP (Session Initiation Protocol). In this work, we bring some benefits to end-user over a regular video conference system which is widely used. These benefits come from features that support remote learning purpose to make teachers and students feel better when dealing with distance learning. The main features include voice and video communications, Power Point layout, PDF reader, interactive white board, chat, file sharing. All such service sessions are established by using SIP. To allow end user to access the system easily, we deploy LDAP (Light-Weight Directory Access Protocol) for this purpose. The system is capable for both IPv4 and IPv6. A system prototype has been developed. We are running such distance learning system between the university campuses for years. We have confirmed the benefits over traditional tools.

Keywords: IDL, SIP, e- learning, distance learning, IP Network

1. Introduction

The Internet has become a high potential platform for teaching and learning. The enormous capacity and connectivity of the Internet for delivering information have fostered a worldwide movement across all education levels to re-examine and evolve the current learning and teaching infrastructure. A number of applications serving such purposes have been developed].

Recently, we have seen a large number of LMS (Learning Management Systems) deployed widely around the world. LMS offers variety of services and managements not only to all participants (for example students, lecturers, academic support staff) but also to administrative works (for example registration, auditing). However, LMS does support non- interactive services. So, we need other kind of tools if we would like to make a real time communication between parties.

Currently, the pervasive deployment of distance learning service is unlikely due to the use of non-standard protocols, as well as restrictive ability in supporting real-time interactions between users. We have found that interactive distance learning (IDL) offers much better facility than a normal video conference (mostly are set-top box, e.g. Avaya, Tanburge, Polycom, Sony). For example, power-point presentations and on-line document sharing are integrated in IDL. These features will encourage ‘good feeling of teaching and learning’ that cannot be found from a regular video conferencing tools.

We believe that interactive distance learning is a better suited platform for distance learning if it can employ the emerging SIP (Session Initiation Protocol) [11], [22] effectively. To that end, we present a development of interactive distance learning based on SIP.

This paper is organised as follows: In the next section, we overview of SIP conferencing architecture. In section 3, we present some functions of SIP deployed by IDL, Section 4 presents a software prototype developed at CNR. In the last section, section 5, we conclude our work.

2. Overview of [1]SIP Conferencing Architecture

As SIP only provides the method for dealing with sessions, it has to work alongside other protocols and standards, for example RTP (Real-time Transport Protocol) [55], SDP (Session Description Protocol) [66].

SIP is a request-then-response protocol. Both requests and responses are well defined as shown below. To facilitate proper communication, SIP messages are exchanged when a call is set up, modified, and terminated. They are simple but yet flexible enough to welcome innovative services for example, instant messaging, presence service and voting.

Request Methods

- INVITE is used to initiate a call and is also used to change call parameters once a session has been established (as a re- INVITE process).
- ACK is an acknowledgement and confirms response to an invite.
- BYE terminates a call in progress.
- CANCEL Terminates the invite and stops all searching
- OPTIONS Used to query the capabilities of a device.
- REGISTER: Registers a device's location with a server.
- INFO Sends information during a session but does not modify the session state.

Response Types

Response messages contain numerical responses and are similar to HTTP format:

- 1xx provisional, searching, ringing, queuing
- 2xx success
- 3xx redirection, forwarding
- 4xx request failure (client)
- 5xx server failures
- 6xx global failure (busy, refusal, not available anywhere

To participate in SIP session, end-points, so called user agents, have to support all of mentioned messages. For one-to-one communication, a SIP server is introduced to ease mechanism of finding another user agent. So, the user agent can be reached without knowing its IP address. A few components such as Focus and Mixer shall be added to achieve conferencing call.

The following are SIP framework and architecture for making a conferencing call [33] [44]. As shown in Figure 1, Conference participants form SIP dialog with the Conference Server in order to exchange conference parameters. Most of parameters are either media related or policy related. Upon agreement of the parameters, participants will exchange their media such as voice or video over RTP.

[1] This work is partly supported by STIC-Asia Project : My SIP: Multimedia Architecture based on SIP

Figure 1 Distance learning system

SIP and RTP are major messages exchanged between participants. A number of SIP messages can be considered to be very light comparing to one of RTP messages. SIP exchanging point is mostly suggested to be centralized. However, RTP exchanging point can be chosen between centralised or distributed. These choices are discussed in 2.1, 2.2 and 2.3.

2.1 Centralised Server

A distance learning is used when each participation remotely communicate to each other. The simplest way of connecting each other together is using a central conference system, as shown in Figure 2, which has been deployed by a telecommunication system for long time. The server receives the media streams from all participants, mixes them and redistributes the appropriate media stream back to the participants. In SIP context, the conference server consists of "focus" which is a conference policy service. For example in case there is a conference moderator, he will be asked first to approve the joining of a new participant.

The system consists of Focus (is SIP Server, used for signaling control), Conference Notification Service (used for multi-conference), Conference Policy (consisting of Membership Policy and Media Policy), Conference Policy Control Protocol (CPCP), Mixer (used for multimedia stream mixer), and conference-aware Participant.

Figure 2 SIP-based central conference architecture

2.2 Endpoint Server

In this case a participant user agent (usually conference initiator) acts as a Focus and mixes the media streams locally, as shown in Figure 3 [4]. This scenario is more suitable for small scale conferences of three or four participants.

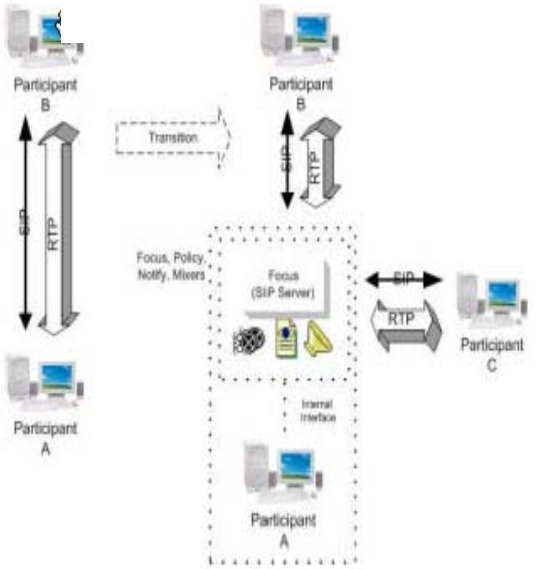

Figure 3 SIP based end-point server

2.3 Distributed Mixing

In this case, there is one focus acting as a central signaling controller. However, mixer is not necessary in the central. Mixers can be placed at any sub-net or even at the end points, as shown in Figure 4. All participants join in a common multicast group. All media streams are sent to such group.

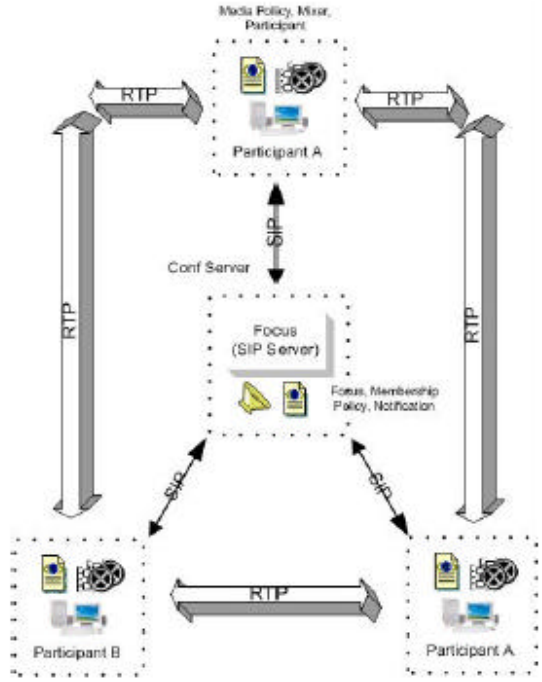

Figure 4 SIP based distributed media mixing architecture

3. SIP-Based Conferencing for IDL

The centralized communication described in IETF is deployed in this project. The components involved in this system are: conference notification server, conference policy server, focus, and media mixer. All service registrations and configurations are done via standard web browser. The following steps are necessary in our design as shown in Figure 55.

1. A new user makes a registration via web browser. In return, login name and password will be given. Registered users can skip this one.

2. A registered user login to the system and creates a conference room. Conference ID will be assigned in this stage.

3. When the conference ID is given, the owner can add conference members (from registered user list) along with their permissions. Roles might be defined prior to this step, so permission assignment can be done easily. One of required role is "Conference Owner" must be assigned to at least one user. This information will be sent to Focus to control a conference policy.

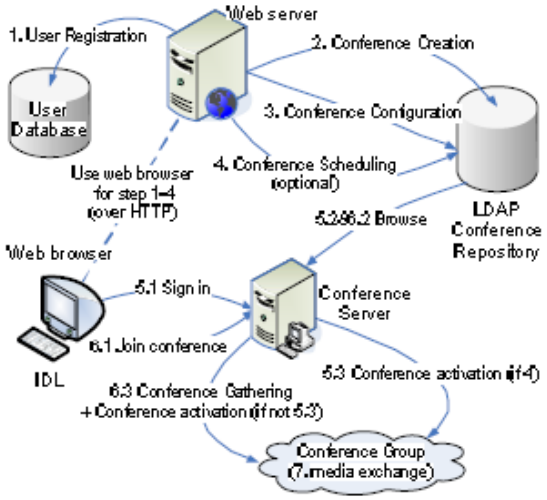

Figure 5 IDL Conference Setup

4. Conference owner can schedule the conference, so participants see that the conference is coming. Although this step is optional, it is recommended.

5. After the conference owner sign in (5.1), any scheduled conference sessions are created corresponding to their policy (5.2, 5.3). Otherwise, the conference owner must join the conference before any authorized participants do. This is to make sure that resources will not be occupied unnecessarily.

6. Once the conference owner join the conference (6.1, 6.2), SIP conference server send INVITE with SIP URI of the conference server to any online conference

members (6.3). Conference setup is very easy in this way.

7. SIP dialogs are then established among participants. Media are exchanged right after SIP session setup is done.

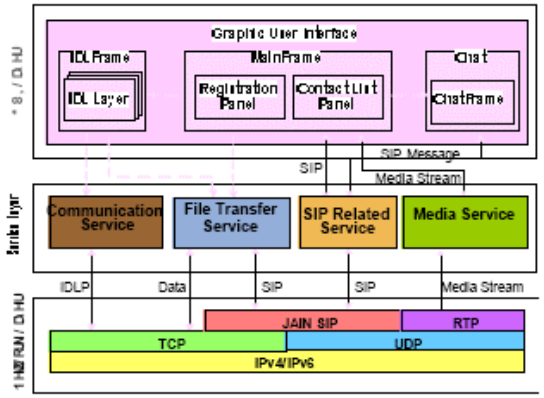

Figure 6 Inside-application IDL architecture

Other than web server and conference server which play very important roles in conference server which play very important role in setup, IDL application is another complex architecture. Its inside-application architecture is briefly introduced in Figure 6. IDL implements various protocol supports in different layers, for example media codec, media specific transport layer, session management protocol and presentation-specific protocol. This is to make sure that IDL is flexible and extensible. The architecture can be categorised into three: GUI Layer, Service Layer and Network Layer. IDLP (IDL Protocol) is an invented XML-based protocol describing presentation action like drawing a line on a whiteboard.

4. A Prototype Development

We use Java to develop our software prototype due to a large variety of API sets. Customized JMF (Java Media Framework) are used to handle voice and video services (media stream). Capability to support better ratio of media quality to connection quality, few more media codecs have been added to our framework, namely MPEG4 and Speex. Speex is also deployed in Skype to offer better voice quality over a congested network. For PowerPoint display and synchronization, we use Jacob API for image transforming

Our system employs LDAP (Light Weight Directory Access Protocol) for two major functionalities. Firstly, participants can be authenticated with the same accounts of other applications. This service is integrated to all Internet access for the whole university (our service is so called "PSU Passport"). Secondly, LDAP server acts like repository for current active conference. SIP server browses the repository to be aware of conferences which it has to create and manage. The following working procedures are used in our system, as shown in Figure 7.

Figure 7 Menu-driven for IDL sessions

1. To activate the IDL, click on a desired subject. The service then calls Call-Servlet with the subject ID and URL. Browser calls SIP-IDL

2. Call-Servlet sends the subject ID seeking IP multicast group which associates to this subject.

3. If students would like to participate, LDAP is used for authorization (with the subject containing registered student IDs).

To eliminate a complicated of using IDL, a menu driven is used. All subjects that need to use IDL for their teaching class can pop-up the IDL and do auto-configuration as pre-describe by an administrator (could be a network administrator plus a registration office, for example).

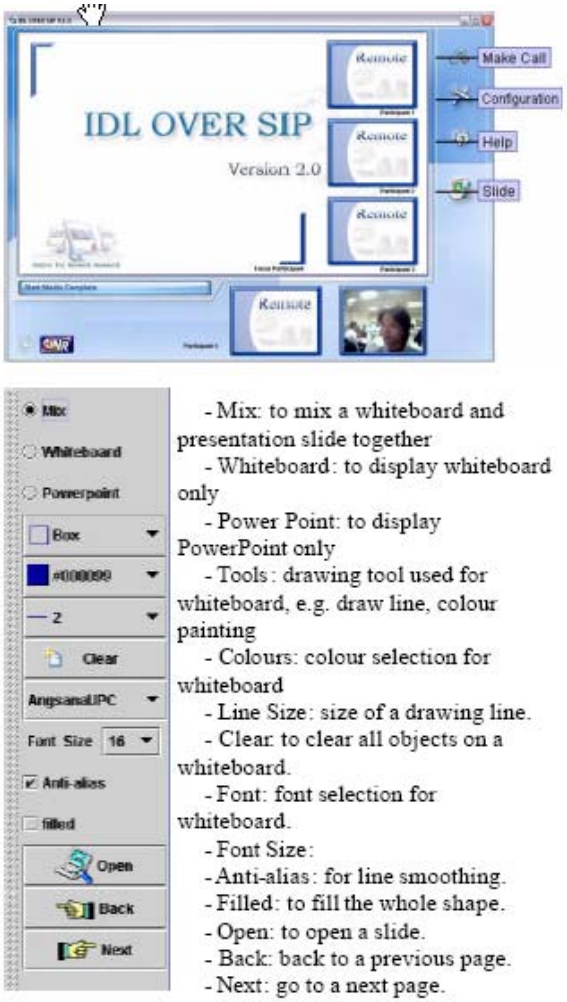

Figure 8 GUI of CNR-IDL

Figure 8 shows our GUI prototype software after calling the IDL. The IDL GUI is so simple. To make a connection, just click on Make Call icon. This is a toggle, click again for hang up the call. We can see all video screens as well as slide presentation can be shown in one page. To make a zoom-in and zoom-out, just click on the objects, e.g. video frame. Such object will be enlarged, click again to reduce it to the original size.

5 Conclusions

The essential session functions of IDL can be relieved through the use of SIP (Session Initiation Protocol), which is the standard internet session protocol emerging in a very fast pace as part of the internet infra-structure service. That means, SIP-based IDLs will be simpler to design and easier to manage. It can take advantage of the useful features of SIP, such as secured session management and mobility (session) management. The IDL offers voice and video communications, Power Point layout, PDF reader, interactive white board, chat, and file sharing. All such service sessions are established by using SIP. To allow end user to access the system easily, we deploy LDAP (Light-Weight Directory Access Protocol) for this purpose. The system is capable for both IPv4 and IPv6. We have demonstrated on our software prototype used daily over a year between PSU campuses. All benefits presented here can be claimed by this deployment easily.

6. References

[1] "Session Initiation Protocol Call Control –Conferencing for User Agents", http://www.ietf.org/proceedings/04aug/I-D/draft-ietf-sipping-cc-conferencing04.txt (07-2004)

[2] "A Session Initiation Protocol (SIP) Event Package for Conference State", http://www.ietf.org/proceedings/04aug/I-D/draft-ietf-sipping-conference-package-05.txt (07-2004)

[3] "A Framework for Conferencing with the Session Initiation Protocol", http://www.ietf.org/proceedings/04aug/I-D/draft-ietf-sipping-conferencingframework-02.txt (07- 2004)

[4] "High Level Requirements for Tightly Coupled SIP Conferencing", http://www.ietf.org/proceedings/04aug/I-D/draft-ietf-sipping-conferencing requirements -01.txt (07-2004)

[5] RFC 1889, A Transport Protocol for Real-Time Applications

[6] RFC 2327, SDP (Session Description Protocol)