A Dynamic Scalable Service Model for SIP-based Video Conference

Zhen YANG, Huadong MA and Ji ZHANG
School of Computer Sci. and Techno., Beijing University of Posts and Telecomm., China
mhd@bupt.edu.cn

Abstract

As an extensible signaling protocol, SIP (Session Initiation Protocol) can be applied in developing video conference system. Traditionally, for SIP-based centralized video conference system, conferencing scale is limited by both the capability of conference server and the availability of bandwidth. The paper focuses on how to keep conferencing when the number of conference users increases. Based on the study of the SIP Protocol, we analyze the drawback of the existing video conference model, and propose a dynamic scalable service model for SIP-based video conference. By this model, the extra service requests can be transferred and served in the cooperated conference servers. The paper presents the conferencing control policy of the model in detail. We developed a prototype of video conference system based on the above model. Experimental results show this service model works well.

Keywords: SIP, Video Conference, Scalability, Service Model.

1. Introduction

The fast development of Internet technologies has provided the basis for multimedia video conference in IP network. With the function of transmitting multimedia information, such as text, graphics, audio and video, video conference breaks the limit of human communications due to the geographical location of participants.

At present, there are two standards to support the development of video conference systems, i.e. ITU-T H.323 [2] and the Session Initiation Protocol (SIP)[1] recommended by the IETF. Because H.323 follows the traditional telephony signal model, it is difficult to control and be extended. However, SIP is flexible and open. Thus, in this paper, SIP is adopted as the basis to study the dynamic scalable video conferencing system.

There are three models for conferencing controls: loosely coupled conference, fully distributed multiparty conferencing and tightly coupled conference [3]. Their common drawback is absence of dynamic extension, i.e., they are not able to dynamically increase the capacity of conference system during the conferencing. To overcome the shortcoming, this paper proposes a dynamic scalable service model for the SIP-based video conference. Under the precondition of no negative effects on the conferencing, the requirements on extending the system's capability can be met by dynamic increasing of conference servers and by transferring the subsequent services to the cooperated conference servers.

The rest of the paper is organized as follows: In section 2, we discuss the related work. Section 3 presents a scalable service model for SIP-based video conferencing architecture. In section 4, we describe the SIP flows of the service model. In section 5, we introduce the implementation of video conference system based on the scalable service model. Finally, we make some concluding remarks in section 6.

2. Related works

2.1. SIP Protocol

SIP [1] was developed by the IETF multiparty multimedia session control working group and has been applied broadly [4,5,6,7,8,9]. SIP is an application-layer control protocol that can establish, modify and terminate multimedia sessions such as Internet conferencing, telephony, events notification and instant messaging. SIP is similar in syntax and grammar to other Internet protocols such as SMTP (Simple Mail Transfer Protocol) used for e-mail and HTTP used for the World Wide Web. The architecture of SIP follows the client-server model, the underlying transport of SIP messages can either be in UDP, TCP or Stream Control Transmission Protocol (SCTP).

However, SIP is not a vertically integrated communications system, and a complete multimedia architecture cannot be built only by using SIP. Usually, SIP is rather a component that can be used with other IETF protocols to build a complete multimedia architecture. Typically, these architectures include protocols such as the Real-time Transport Protocol (RTP) [10] for transporting real-time data and providing QoS feedback, the Real-Time Streaming Protocol (RTSP) [11] for controlling delivery of

The work reported in this paper is supported by the NSFC under Grant 60242002 and the NCET Program of MOE, China

streaming media, and the Session Description Protocol (SDP) [12] which is used to describe multimedia sessions.

There are six request methods mainly defined in SIP: INVITE, ACK, BYE, OPTION, CANCEL and REGISTER. Certainly, SIP has also been extended to generate event notifications and instant messages [13].

2.2. SIP-based video conference system

Efforts to implement SIP features for video conference system have been going on for the past few years [6,8,9,14,15,16,17,18]. Typically, there are three models for conferencing control: loosely coupled conference, fully distributed multiparty conference and tightly coupled conference [3]. Due to the absence of a widely available multicast service on Internet at present, the model of our implementation is based on the currently more common centralized framework and tightly controlled session model. IETF describes a framework in some drafts ([3, 14, 15, 16]) for how such SIP-based tightly coupled conferences can be held, including the overall architecture, protocol components needed for multi-party conferencing and the flows of the conference. The IETF framework provides us with the research basis. However, IETF framework has a drawback: there is only one conference server acting as central control unit, whose conference service capability limits the scale of conference, so that the conference lacks of extensibility. We overcome the shortcoming by dynamically increasing the number of conference servers and transferring the subsequent service to the increased conference server without stopping the conference. We adopt the object oriented method to manage conferences, which means the realization of conference dynamic extension depends on the accession of the members under the case that each conference server joining the conference becomes one member of the conference objects. We also define some special conference policies that embody the above idea and these policies can be implemented by SIP.

3. The scalable service model for SIP-based video conference

As shown in Figure 1, the scalable service model for SIP-based video conference system consists of three kinds of SIP entities: User Agent (UA), Conference Server (CS) and Manager Server (MS). The UA and CS are called by a joint name, conference member. In subsection 3.1, we introduce conference members and their basic functions. Subsection 3.2 defines the concept of conference object and discusses its intra-structure. In subsection 3.3, we describe the structure of manager server and its functions, and present the conference control policy in detail in subsection 3.4.

3.1. Conference member

The model includes two kinds of conference members: UA and CS. Conference members are the basic elements of conference and act as the participants and the entities of implementation.

User Agent (UA): Each UA consists of a User Agent Client (UAC) and a User Agent Server (UAS). A UAC is a logical entity that creates a new request, and then uses the client transaction state machinery to send it. A UAS is also a logical entity that generates a response to a SIP request. In this paper, a UA also represents a client of video conference system.

Conference Server (CS): The main functions of CS are as follows: (1) SIP signal interaction with UAs and MS; (2) receiving, handling and redistributing the media streams. Each CS is not only a logical entity, but also corresponding to a physical entity, e.g., a computer.

3.2. Conference object

As a logic entity, a conference object, created and managed by MS, creates and manages the conference members of the conference. The relationship between conference and conference object is one-to-one.

As shown in Figure 2, there are two tables in the conference object: User Agent Table (UAT) and Conference Server Table (CST). The two tables list the related information of UAs and CSs in the conference system, respectively. The virtual line linking the two tables means that one user has logged into some conference server. For example, the virtual line linking UA1 to CS1 indicates the UA1 has logged into the CS1. Furthermore, we note that the Resource Request Module (RRM), as a part of the conference object, is devoted to checking the load status of every CS existing in the CST and requesting new CS to join the conference.

3.3. Manager server

In the model, we design a single MS which collectively implements the functions of register server, redirect server and proxy server. In addition, the additional functions of MS are as follows: (1) creating and managing the conference object, (2) resource management and scheduling.

The MS is composed of five modules: Communication Interface (CI), Conference Object Queue (COQ), Resource Scheduler (RS), User Agent Management Table (UAMT) and Conference Server Management Table (CSMT) (see Figure 1).

CI takes charge of the MS communication with CSs and UAs. Creating, managing and deleting the conference object are the responsibilities taken by COQ. RS services resource management and scheduling. The

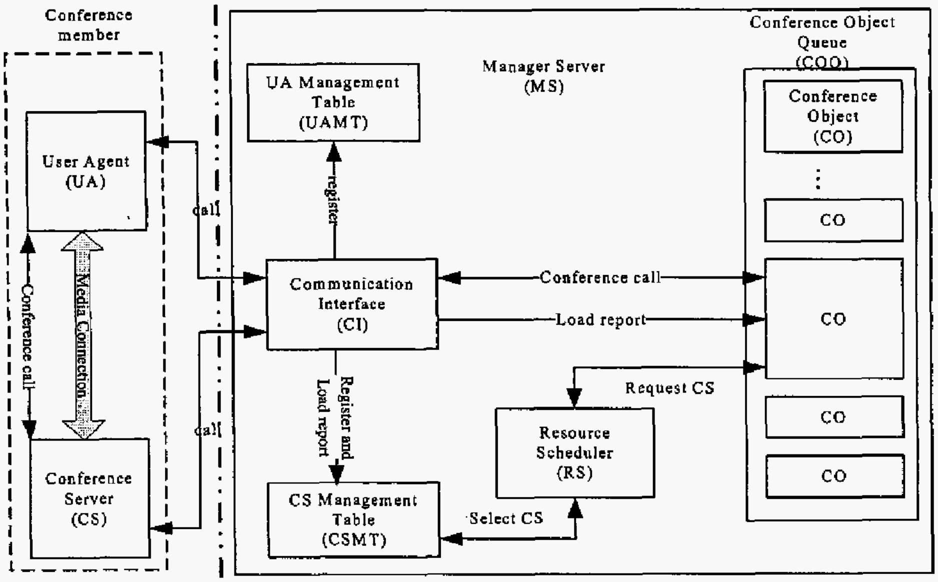

Figure 1. The scalable service model for SIP-based video conference

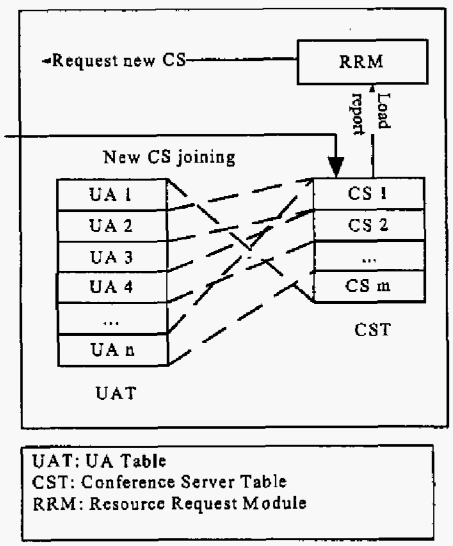

Figure 2. The internal structure of conference object

UAMT and CSMT manage the information of all the registered CSs and UAs. The UAT and CST of each conference object are regarded as subsets of UAMT and CSMT, respectively.

3.4. Conference control policies

At first, we introduce the conception of load ratio. The load ratio is the performance index of CS. In the paper, we take the resource utilizations of CS as the load ratio of CS because it shows an approximately linear positive correlation between the resource utilization and the load ratio.

In order to implement the dynamic scalability, we make some conference control policies for the model.

(a) Once the UA or CS logs into the network, it registers into the MS immediately. The register information of UAs and CSs are stored into the UAMT and CSMT, respectively. As a CS registering into the MS successfully, the MS asks the CS to report its load ratio every a certain time (e.g. 5s). The load ratio is written into the CSMT and is also stored into the CST of a conference if the CS has joined the conference.

(b) Before a conference holding, a conference object will be created and inserted into COQ by MS, and the RRM of the conference object will request the RS to select a CS as the initial CS of the conference. Then, the conference object will copy the relative item from the CSMT to its CST. At the same time, the MS invites the CS to join the conference.

(c) UA will send a request to MS for the CS address, if it expects to join a conference. After the MS receives the request, the UAT of the conference object will copy the relative item from the UAMT. Then, the RRM chooses the CS with the least load among all CSs of the conference to interact the conference signals with the UA. Finally, the conference object will build up the relationship between the UA and CS. Once a UA exits a conference or fails to log into CS, the conference object will cancel the relationship between the UA and CS and delete the related items of the UA from the UAT.

(d) During a conference, the RRM of the conference object will check the load status of every CS in CST. Once the load of each CS of the conference reaches the utmost, the RRM will send a request for a new CS to the RS. Then the RS will select CS with the least load among those who haven't joined the conference from the CSMT, and then the selected CS will be inserted into the CST. At the same time, the MS sends a request to invite the CS to join the conference. When this step is accomplished, the service requests of the subsequent UAs will be transferred to the CS according to (c).

(e) When a conference is over, the MS will send a message to CS and delete the conference object corresponding to the conference from the COQ.

4. The SIP flows of the model

In this section, we will look at the SIP flows of some interesting phases in the conference, where the SIP flows conform to the conference control policies of the model.

4.1. The UA flows

When a user (UA) using any terminal logs on to a network, the REGISTER messages are sent to the MS. These messages record the related information, such as address and URI, are stored into the UAMT (see the conference control policy (a)).

The Figure 3 depicts the SIP flows of UA entering and exiting a conference, which follow the conference control policy (c). The flow of UA joining the conference is as follow. First, after receiving the INVITE request sent by UA for getting the address of CS, MS responses the ADDRESS [19] message which includes the information about the address of CS, and then UA returns ACK to MS. Second, UA joins the conference as shown in step 4, 5, 6 of the Figure 3. Finally, the UA sends MS such multiple SUBSCRIBE requests as to ask to download the participants information of the conference, or to ask to get the audio streams or video streams of some users. The SUBSCRIBE request contains an "event" header indicating the type of event the UA is subscribing to and an "expires" header specifying the duration of the subscription. The request can be refreshed whenever the subscription has expired. If a subscriber wants to unsubscribe, it can send a SUBSCRIBE message with an expiration time of zero. The SUBSCRIBE (event = download) request asking to download the participant list of the conference will be sent to the MS as soon as a UA joins the conference. After receiving this request, MS will intermittently (e.g. 5s) send the NOTIFY response to the UA in order to inform it the participant list. At the same time, this response can also be triggered when other UAs enter or exit. After joining a conference, the UA will send media streams to the CS on which it has logged. Then, the UA will receive media streams from the logged CS or other CSs if it requests the media streams of other UAs which have logged on the other CSs.

Furthermore, there are other SIP flows of UA such as a UA inviting another UA to join the conference. For example, if UA1, as a conference user, expects to invite UA2 to join the conference, it will send REFER [20] to MS which will select the least load CS_i among all CSs of the conference. The MS will subsequently send MESSAGE [21] to CS_i which will send INVITE signal to UA2, in order to invite UA2 to entry the conference.

When a UA exits the conference, it will send BYE message to MS and CS.

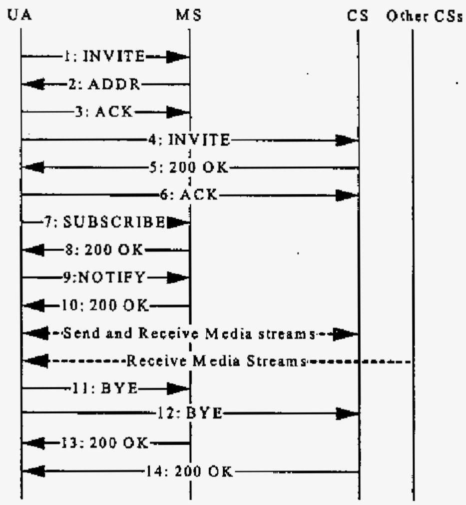

Figure 3. The basic SIP flows of UA

4.2. The CS flows

The CS will send REGISTER message to MS as soon as it logs on the network. After receiving the first REGISTER from CS, MS will send SUBSCRIBE (event = query) for checking the CS load ratio. At the same time, CS intermittently (e.g. 5s) responds NOTIFY message to MS reporting its load situation. This step follows the conference control policy (a).

The SIP flow of CS dynamically joining a conference is as follows. According to the conference control policy (d), when each conference CS has overloaded, the MS will send INVITE message to the least load CS from the CSMT. Then the selected CS will respond 200 OK to the MS, if it agrees on joining the conference.

When the conference is over, the MS will send BYE message to the CS which responses 200 OK to the MS. After having received the response, the MS will delete the conference object according to the conference control policy (e).

5. A prototype system and performance evaluation

According to our model, we designed and implemented a SIP-based video conference system. The development of the system used the multimedia middleware proposed in [8,9]. The interfaces of UA, CS, MS are shown in Figure 4, 5 and 6, respectively.

In the system, each UA (per participant during a video conferencing) executes on a separate host. We used 50 UAs, one MS and two CSs (CS1 and CS2) to evaluate the performance of our implementation under the precondition that there is only one conference existed.

Initially there was only CS1 joining the conference and the number of UAs was increased one by one. We measured the load ratio of CS1 and CS2 every 5s. According to the discussion above, the load ratio of CS was recorded in the CSMT, and if the CS had joined the conference, it was also recorded in the CST of the conference object.

In Figure 7, we show the load ratios of CS1 and CS2 as each user joins the conference. As the number of UAs increases from 0 to 33, the load ratio of CS1 increases from 0 to 82% approximately linearly, while that of CS2 still keeps 0. However, if the load ratio of CS1 is over 80%, and the subsequent UAs keep on logging into CS1, the resource of CS1 would be exhausted and the conference would end abnormally. To avoid the emergent situation, the MS corresponsively sends INVITE request to CS2 and invites it to join the conference acting as the receiver for the sequent service requests. As the number of UAs increases from 33 to 50, the load ratio of CS1 keeps under 90%, while that of CS2 increases from 0 to 58%.

The above results show that our service model is characterized with the dynamic scalability without any negative effects on the conference.

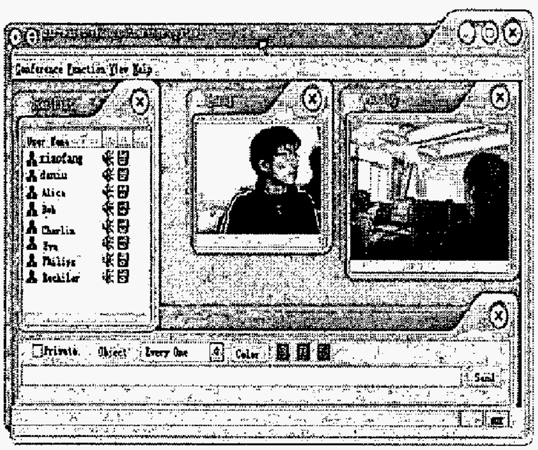

Figure 4. The interface of UA

6. Conclusion

This paper provides a service model to support dynamic scalability of SIP-based video conference service. Based on the study of the SIP Protocol, we analyzed the drawback of the existing video conference models, and proposed a dynamic scalable service model for SIP-based video conference system. We also presented the structure, the control policies and SIP signal flows of the model. By this model, the extra service requests can be transferred and served in the cooperated conference servers. Furthermore, we implemented a prototype system based on the model, and the experimental results show the service model works well.

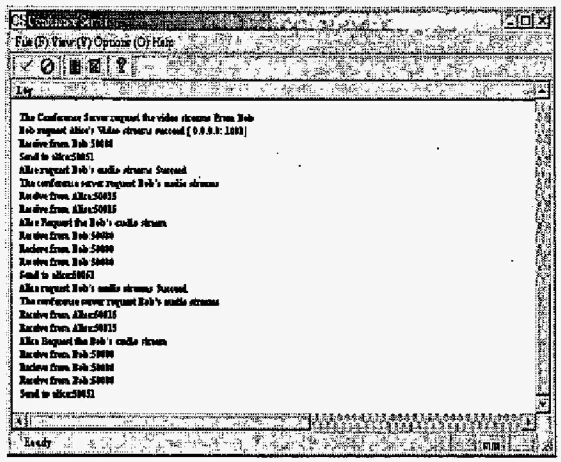

Figure 5. The interface of CS

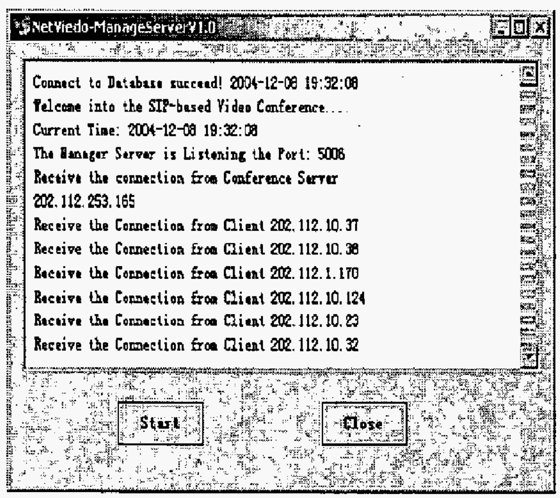

Figure 6. The interface of MS

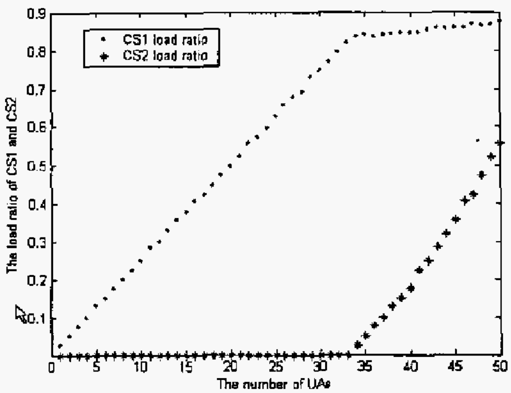

Figure 7. The load ratio of CS1 and CS2 vs the number of UAs

References

[1] J. Rosenberg, H. Schulzrinne et al., SIP: Session Initiation Protocol, *IETF RFC 3261*, June 2002.

[2] ITU-T Rec. H.323, Packet based Multimedia Communications Systems, *Telecommunication Standardization Sector of ITU*, July 2003.

[3] J. Rosenberg, A Framework for Conferencing with the Session Initiation Protocol, *IETF Draft*, Oct.2004.

[4] 3rd Generation Partnership Project, IP Multimedia Subsystem (IMS), Stage 2(Release 5), 3GPP TS 23.228 V5.1.0, July 2000.

[5] A. Meddahi, G. Vanwormhoudt, SIP for e-Learning Services, *10th International Conference on Communication Technology (ICT 2003)*, vol.1, pp.522–529, 2003.

[6] R. Venkatesha Prasad, Richard Hurni, H S Jamadagni, A Scalable Distributed VoIP Conferencing Using SIP, *Proceedings of the Eighth IEEE International Symposium on Computers and Communication*, vol.1, pp.608-613, 2003.

[7] Knarig Arabshian and Henning Schulzrinne, A SIP-based Medical Event Monitoring System, *MOBIQUITOUS 2004*, pp.319-324, 2004.

[8] Ji Zhang, Huadong MA, An Approach to Developing Multimedia Services Based On SIP, *Proceedings of ICCC2004*, Vol.1, pp95-101, Beijing, Sept. 2004.

[9] Huadong MA, Ji Zhang, A Unified Framework of Multimedia Services Based on SIP, *Proceedings of 2005 IEEE Conference on Consumer Electronics*, pp389-390, Las Vegas, Jan. 2005.

[10] H. Schulzrinne, et al., RTP: a transport protocol for real-time applications, *IETF RFC 1889*, Jan. 1996.

[11] Schulzrinne, H., Rao, R. and R. Lanphier, Real Time Streaming Protocol (RTSP), *IETF RFC 2326*, April 1998.

[12] Handley, M. and V. Jacobson, SDP: Session Description Protocol, *IETF RFC 2327*, April 1998.

[13] A. B. Roach, Session Initiation Protocol (SIP)-Specific Event Notification, *IETF RFC 3265*, June 2002.

[14] J. Rosenberg, A Session Initiation Protocol (SIP) Event Package for Conference State, *IETF Draft*, July.2004.

[15] A. Johnston, Session Initiation Protocol Call Control - Conferencing for User Agents, *IETF Draft*, July.2004.

[16] M. Handley, J Crowcroft, et at., The internet Multimedia conferencing architecture, *IETF Draft*, 2000.

[17] P. Koskelainen, H. Schulzrinne, et at., A SIP-based conference control framework. Proc. *Nossdav'02*, Miami Beach, Florida, USA. May 2002.

[18] J. Rosenberg, H. Schulzrinne, Models for multiparty conferencing in SIP, *IETF Draft*, 2002.

[19] S. Zeadally, F. Siddiqui, Design and Implementation of a SIP-based VoIP *Architecture, Proceedings of AINA 2004*, vol.2, pp.187-190, 2004.

[20] Sparks, R., The Session Initiation Protocol (SIP) Refer Method, *IETF RFC 3515*, April 2003.

[21] James M. Polk, Requirements for Session Initiation Protocol Location Conveyance, *IETF Draft*, July 2004.