A Scalable Distributed Architecture for Multi-Party Conferencing using SIP

Young-Hoon Cho[1], Moon-Sang Jeong[2] and Jong-Tae Park[2]

[1] Department of Information and Communication, Kyungpook National University
[2] School of Electronic and Electrical Engineering, Kyungpook National University
1370, Sankyuk-Dong, Buk-Gu, Taegu, Korea 702-701
{yhcho,msjeong,jtpark}@ee.knu.ac.kr

Abstract. As various multimedia communication services are increasingly required by Internet users, several signaling protocols have been proposed for the efficient control of complex multimedia communication services. However, the model and architecture of multi-party conferencing which is currently being standardized by IETF has some limitation in scalability to meet the requirement for the management of large-scale multimedia conferencing service. In this article, we have presented a new scalable distributed architecture for the efficient management of large-scale multimedia conferencing service which is based on SIP. The high scalability is achieved by adding, deleting and modifying the multiple mixers and composing conference server network in a distributed way, in a real-time, and without disruption of services. The SIP-based control mechanism for achieving the scalability has been designed in detail. Finally, the performance of the proposed architecture has been evaluated by simulation.

1 Introduction

Internet telephony services provide not only traditional voice services, but also application services based on packet, so these are applied in various multimedia communication services such as video, and multi-party conferencing. Demands for Internet telephony services are steadily increasing. Additionally, signaling protocols, applications, and models have been developed for efficient control of complex multimedia communication services.

Internet telephony services use IETF standard protocols such as H.323, SIP, and MGCP for call control and signaling. An ITU-T's H.323 defines the terminal and other components to provide multimedia communication on a packet network [1]. An SIP (Session Initiation Protocol) is an application layer signaling protocol standardized by IETF which defines initiation, modification, and termination of multimedia communication session between users [2]. Because SIP supports flexibility, user mobility, and various merits, its application area is wide.

Currently, a great deal of research about conferencing models and controlling mechanisms using SIP are being conducted to provide complex multi-party

conferencing. IETF MMUSIC working group studies the general requirements of Internet multimedia conference structure for supporting multi-party conferencing [3]. IETF SIPPING working group proposes several drafts for multi-party conferencing based on SIP [4]. Stream processing like mixing and encoding about various types of media, and performance evaluation are studied in a conference based on centralized server model [8]. Based on the handling method for signaling and redistribution of stream, a conferencing model can be generally divided into end system mixing, multicast, centralized server, and full mesh and the characteristics of each models are described as shown in Table 1.

Table 1. Characteristics of conferencing models

Model	Signaling	Media	Inviting	Joining	Scalability
End System Mixing	Tree	Tree	INVITE	INVITE	Small
Multicast	Pairs	Multicast	INVITE	Multicast Join	Large
Centralized Server	Star	Star	REFER	INVITE	Medium
Full Mesh	Star	Full Mesh	REFER+ Server msg.	INVITE+ Server msg.	Medium

Even though multicast model has huge scalability, but it is hard to apply because multicast is not deployed widely. Currently, the centralized server model is adopted as basic multi-party conferencing model. However, this model has limitations of scalability such as triangular transmission by using single conference server, bottleneck by traffic concentration, and processing overload. Thus, a new scalable conferencing model supporting large scale multi-party conference is needed. More specifically, a new model needs to be designed which can facilitate the transition to the new model and the integration with the existing conferencing model, using standard signaling method. In this paper, we suggest a new scalable distributed architecture for multi-party conferencing using SIP that can reduce traffic and processing load, and that can support scalability by constructing a special network for conferencing servers.

In this architecture, a participant host acquires information of the adjacent conferencing server, and several conferencing servers can join a conference using this information. Stream can be delivered efficiently in this architecture. A distributed conference can be constructed from the existing conferencing model by using the standard signaling procedure, and can distribute load by constructing a conference server network without changing the end host. Using adjacent conference server information and data distribution mechanism, it can provide a virtual multicast conference.

In this paper, we analyze the features of the existing multi-party conferencing models using SIP, and propose a new distributed architecture for multi-party conferencing which can support scalability, load distribution, and traffic distribution. Specifically, we describe a signaling procedure and conferencing mechanism that can make this architecture more scalable.

2 Distributed Conferencing Architecture

In the case of a centralized conferencing model, there are some problems due to the conference control mechanism using single server. First, according to server location, although the conference server is far away from the participants, data transmission between participants is always performed through the conference server. Bottleneck may occur because the traffic of all participants may be concentrated to the server, and the processing load of the server can increase rapidly because the server must mix and encode all the streams. A centralized conferencing model has limitation of scalability in a large scale conference environment. Therefore, a new multi-party conferencing model that can deliver stream effectively and can provide scalability is needed. A new distributed conferencing model which can provide scalability is thus proposed.

We propose distributed conferencing architecture for the large-scale multimedia conferencing service. In Fig. 1, we describe the distributed conferencing architecture which vertically consists of three tiers: a conference management tier, a mixer tier for multimedia stream processing, and the participants. The salient feature of the architecture is that the conference management tier is configured in a distributed way.

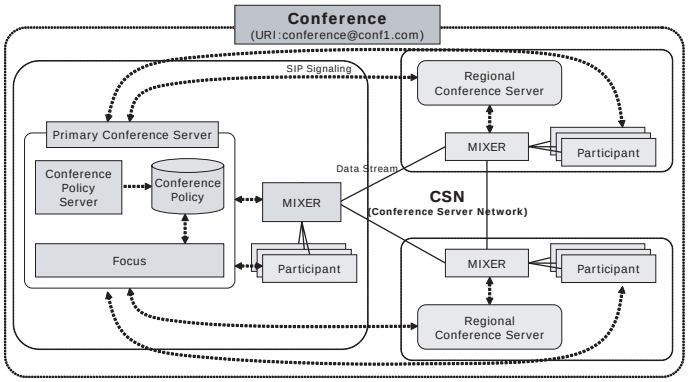

Fig. 1. Distributed Conferencing Architecture

In the distributed conferencing architecture, a conference consists of several local conference servers (CS). Each local conference server contains a focus which is responsible for the management of the corresponding local conferencing service. The focus also manages the corresponding mixers in the region for load sharing and media streaming. In the architecture, one of the conference servers is designated as a primary conference server (PCS). The conference is horizontally comprised of PCS, several regional CSs for signaling and streaming of the conference. Both the PCS and the regional CS control the conferencing operations using the SIP signaling with some extension in accordance with the

conference policy. The CS also handles mixing and redistribution of multimedia streams such as conference video and audio streams. The set of CSs involved in the conference constitute a network called as a conference server network (CSN).

The PCS is responsible for the control of the whole conference in an integrated way. It sets up the CSN and modifies the CSN. It also controls the access to the conference server so that the participants should first get the permission from the PCS to participate in a specific conference session. The PCS announces the conference session information using session announcement protocol (SAP) [12], and handles participation requests. The PCS can add and delete mixers according to the scale of conference, so that it can compose the CSN properly. Thus, the control and mixing operations are distributed in the proposed distributed conferencing architecture, so that the processing overload and traffic concentration can be reduced. These features can greatly enhance the scalability of the conferencing system.

Basically, one of the CSs in a conference is selected automatically to play the role of PCS. If the PCS leaves the CSN, a PCS transition procedure occurs so that another CS in the CSN can be the PCS. Through this, conferencing can be maintained without unnecessary CS. Since the CSN can be configured independently of participants, participants don't have to take care of the composition of the CSN. This makes the signaling and stream procedure of the centralized conferencing model to be used without modification. Additionally, the triangular transmission which is caused by accessing the remote server can be eliminated, and delay and traffic in the core network can be reduced accordingly.

In order to efficiently support the conferencing operation of the proposed distributed architecture, we have extended the SIP signaling method for the exchange of ACS information between the primary focus and the participants. This can be achieved by using ACSInfo header. ACSInfo header is newly defined extended SIP signaling information for the proposed distributed conferencing architecture. The primary focus uses the ACSInfo header information to configure a conference mixer network. The format of the ACSInfo header which is defined in the distributed conferencing model is show below.

```
ACSInfo: SIP-URI or hostport
```

However, even if an end host does not send ACS information, a CS can handle conference composition and signaling process. That is, if the end host is a 'Conference Unaware UA', it can participate in a conference.

A NOTIFY message is used to exchange the information of the participants when the conference status is changed. When participants join a conference, participants subscribe by using the conference notification message, and receive a notification message which contains the current conference status such as participant status, conference server status, PCS transition notification and CS table exchange. Using the participant status notification message, the PCS broadcasts the status change information to all the participants when a participant joins and leaves. The CS status message includes the information on the configuration of CSN and stream transmission mechanism. The CS table exchange message includes the participant list and the stream type of the mixer.

3 Signaling of Distributed Conferencing Architecture

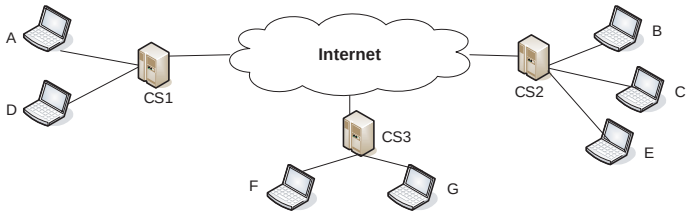

Fig. 2. Test Network for Distributed Conferencing

A distributed conferencing uses SIP signaling to construct a conference and CSN. A conference server operates like a general SIP UA, and a user can join a conference by submitting a connection request to the CS. Figure 2 shows a test network that is designed to examine conference composition procedures of the distributed conferencing. Procedures of invitation, joining, and leaving a test network, changing conference session, and making a PCS transition are examined.

• **Conference Initiation**

Because a distributed conferencing is based on the dial-in conferencing model, the signaling procedure for invitation is to the same as the dial-in model. When host A invites host B, a CS1 that is the ACS of A takes charge of the current conference signaling. Host A requests the creation of a session to the CS1 and then CS1 becomes the PCS for additional signaling and conference control. Finally, host A and B are connected with the CS1 according to the test network.

• **Inviting and Joining**

After a conference is created, if a participant wishes to invite a new user, the participant sends a REFER message to the relevant host. Then, the end host can join a conference by sending an INVITE message to the CS. If a end host wishes to join a conference, the host can acquire the session information about the conference by SAP and access the CS using the INVITE message. At this time, access point is always the PCS. The PCS connects with the current CS using the INVTE message of the participant, or if necessary, handles all signaling procedures after the CSN is composed by adding CS.

If the ACS information of a new participant is equal to the ACS information of existent participants in a single CS conference, it is better to change for efficient stream flowing. For example, when host A invites host C and where host B and C have the same ACS information, the CS transition is occurred. The CS transition procedure is identical to PCS transition procedure.

When a new participant joins a conference, the new CS can participate in the conference according to the CS policy. At this time, a new conference session

is formed between the conference servers. If the PCS decides to participate in a new CS, the PCS sends an INVTE message to the new CS. The CSN initiates a conference session, and the conferencing model of the CSN can be a full mesh, centralized, end system mixing, or hybrid type according to the conference's policy, If the CSN is established and a new CS is added, participants who have the new CS as ACS reestablish their session. For this, a PCS requires relevant participants to reconnect with the new CS sending a REFER message. Accordingly, relevant participants send an INVITE message to the new CS and terminate the session with the old CS.

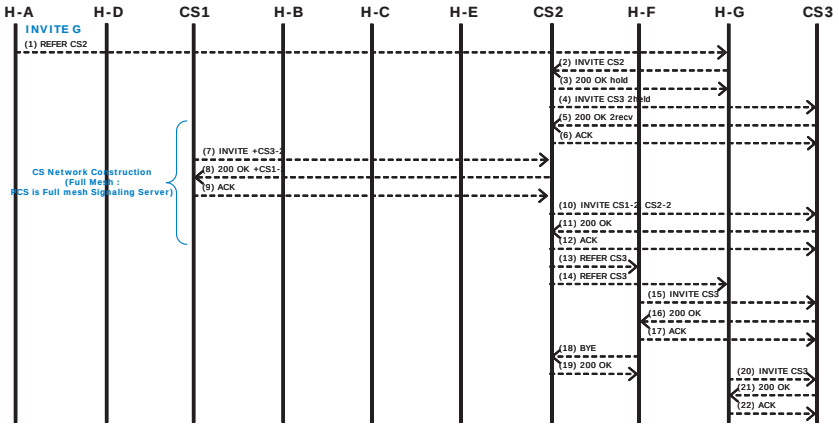

Fig. 3. Signaling Procedures for Joining and CSN Composition

Fig. 3 shows the signaling procedure for inviting a host G where hosts A and D are connected with a CS1, and host B, C, E, and F are connected with CS2. Because hosts F and G have CS3 as their ACS, a CS transition and a CSN re-composition will take place.

• **Leaving**

When a participant wishes to leave a conference session while conference progresses, the participant sends a BYE message to the connected CS. The participant's leaving can leads to CS's leaving in CSN. When a participant leaves, if the number of participants belonging to a CS is lower than the value of minimum CS participants decided by policy, the CS leaves the CSN. Therefore, CS send REFER message to participants belonging to itself that participants will reestablish connection with another CS. When all of remaining participants leave CS, the CS sends a BYE message to the PCS and leaves the CSN. If the CS is a PCS, a PCS transition occurs. The PCS passes the role of PCS to one of the remaining CS, and notifies all participant of the PCS change by sending a NOTIFY message to everyone.

4 Performance Analysis

In this section, we evaluated the performance of the distributed multi-party conferencing model. In particular, we compare the performance of the existing centralized conferencing model with the distributed model that we have proposed. We have measured a signaling delay, a stream transmission delay and a processing load of a conference server for the test network which is shown in Fig. 2. In the centralized mode of conferencing management, the CS1 play the role of the centralized server, whereas in a distributed model, the management operations and the stream distribution is performed by CS1, CS2 and CS3 in a distributed way. The ACS of the nodes A and D is CS1, and those of the nodes B, C, E and F, G are CS2 and CS3, respectively.

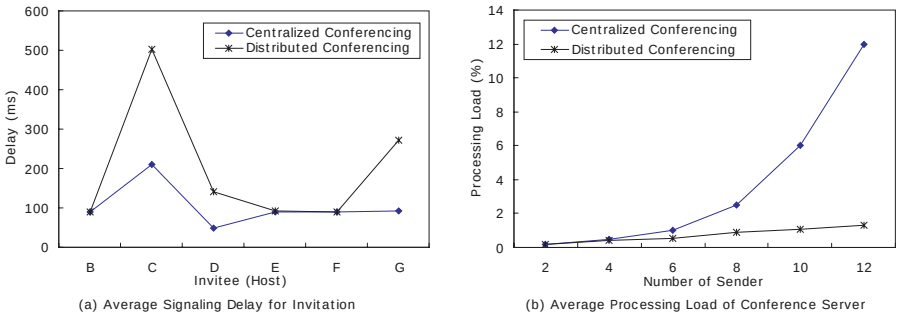

Fig. 4. Performance Analysis for the Distributed Conferencing Model

Fig. 4 (a) shows the delay characteristics when the participant A invites other participants B, C, D, E, F and G. The delay is measured in the average signaling completion time. Initially, the participant A invites the participant B, and in this case, the average signaling delay is identical in both distributed and centralized models. This is indicated by the delay due to inviting the participant B. However, when the participant C is invited, the CSN is re-configured. A new conference server CS2 is joined to handle the media requests from the new participant C, and the existing participant B should be assigned to the CS2. This re-adjustment generates some delay so that the invitation completion time for the participant C is large as shown in Fig. 4 (a). This pattern of re-configuration continues to invite other participants D, E, F, and G, and the related signaling delay times are shown. As shown in Figure 4, the distributed conferencing model creates larger delay time than that of the centralized conferencing model. However, when the number of participants is large, and they are grouped and located in different regions, the signaling delay can be reduced since the regional CS can perform the conference management functions which are related to the corresponding region.

Fig. 4 (b) shows the result of measuring the processing load for encoding/decoding and mixing at the conference server for the transmission of stream.

Both the processing load of the centralized conferencing model and that of the distributed conferencing model are shown in comparison. As shown in Fig. 4 (b), the processing load of the centralized model drastically increases as the number of participants increase, while in the distributed model, the processing load is almost constant. This illustrates the fact that the distributed model performs n better than the centralized model with regard to the scalability.

In the case of a large conference, because a number of hosts connected to each CS are smaller than the centralized conferencing, a processing load of each CS is lower than the centralize server. Especially, if a CSN may be constructed a tree topology instead of a full mesh topology in a test network, a load of CSs may be more decreased.

5 Conclusion

In this paper, we suggest a new distributed conferencing architecture which can provide better scalability that appears to be very important feature in a wide-scale Internet community. We specifically design signaling procedures and conferencing mechanisms for this architecture. The proposed architecture facilitates both the integration with the existing models and transition from the existing models, providing efficient load and traffic distribution, thereby achieving great scalability. For further study, we are planning to apply the architecture in a real environment and to evaluate the performance enhancement by comparison with other works.

References

1. ITU-T "Recommendation H.323: Packet-based Multimedia Communications Systems", ITU, Geneva, November 2000
2. J. Rosenberg, H. Schulzrinne, Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
3. M. Handley, J. Crowcroft, C. Bormann and J. Ott, "The Internet Multimedia Conferencing Architecture", Internet-Draft, MMUSIC WG, July 2000
4. J. Rosenberg and H. Schulzrinne, "Models for Multi Party Conferencing in SIP", Internet-Draft, SIPPING WG, July 2003
5. O. Levin and R.Even, "High Level Requirements for Tightly Coupled SIP Conferencing", Internet-Draft, SIPPING WG, April 2003
6. J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol", Internet-Draft, SIPPING WG, May 2003
7. A. Johnston and O. Levin, "Session Initiation Protocol Call Control - Conferencing for User Agents", Internet-Draft, SIPPING WG, June 2003
8. Kundan Singh, Gautam Nair and Henning Schulzrinne, "Centralized Conferencing using SIP", Proceedings of the 2nd IP-Telephony Workshop, April 2001
9. J. Veizades, E. Guttman, C. Perkins and S. Kaplan, "Service Location Protocol", RFC 2165, June 1997
10. M. Handley, C. Perkins and E. Whelan, "Service Announcement Protocol", RFX 2974, October 2000