

# Performance Evaluation of an XML-Based Conference Control Protocol for Centralized VoIP Conference

Eunsook Kim, Jong-Hwa Yi, Shin-Gak Kang, and Kishik Park

*ABSTRACT*—For communication between heterogeneous conference systems, it is important to build a conference control protocol independent from signaling protocols. We simulate an XML-based conference control that is under consideration as a standard mechanism. We describe the framework and operations for easy implementation in heterogeneous conference systems. The simulation results show that the proposed control protocol provides a consistent service for an increasing number of conferences and participants in small to mid-size centralized conferencing.

*Keywords*—Internet conference, SIP, XML.

## I. Introduction

Conference control is one of the essential functions of multimedia conference systems. It coordinates and manages multiple users using different media, operating systems and conference applications. It also provides the basic operations of conferencing.

Previous studies on this topic focused on multicast conferencing [1], [2] and are not adaptable to centralized conferencing, which is of main consideration these days.

Standardization efforts have been made for building conference control protocols. ITU-T developed conference control protocols as a part of the H.323 [3] series of recommendations, T.120 [4].

However, T.120 has limited scalability with a lack of key features for conference control.

The Internet Engineering Task Force (IETF) SIPping Working Group (WG) also has been making efforts to build conference control mechanisms by session initiation protocol (SIP) [5]. However, SIP itself does not offer to deliver conference control information. Thus, IETF organized XCON BoF in July 2003 and started to develop a standardized suite of conference control protocols for tightly-coupled multimedia conferences. However, we are at an early stage where only the basic control protocol requirements and framework have been published [6], [7].

Therefore, it is important to build an efficient conference control protocol for SIP-based conference systems. Furthermore, it would be a good idea to build the protocol independent from the signaling protocols in order to support H.323-based conference systems.

For this reason, we propose an XML-based conference control protocol (XCCP). To confirm to IETF recommendations, XCCP conference policy data and operations are defined by XML, which is independent of the platforms or devices used [6], [7]. This paper explains the framework of the designed protocol we design and shows the simulation results by prototyping the basic operations of the XCCP.

The simulation results show that the XCCP provides a stable service for small to mid-sized conferences. We believe that these results give the kind of quality information needed for deciding upon standardized mechanisms for centralized conference control.

This paper is organized as follows. Section II describes the

---

Manuscript received Oct. 14, 2003.

Eunsook Kim (phone: +82 42 850 6124, email: eunah@etri.re.kr), Jong-Hwa Yi (email: jhyice@etri.re.kr), and Shin-Gak Kang (email: sgkang@etri.re.kr) are with Basic Research Laboratory, ETRI, Daejeon, Korea.

Kishik Park (email: kipark@etri.re.kr) is with IT Service Research Division, ETRI, Daejeon, Korea.

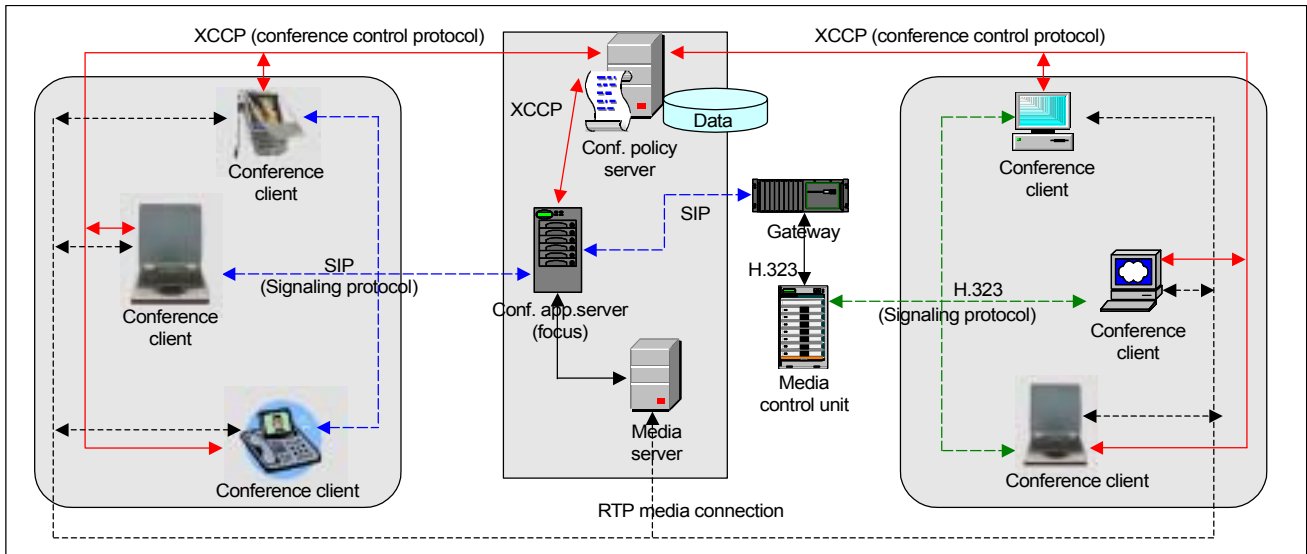


Fig. 1. Overall conference system architecture.

framework of the XCCP, while section III explains its operations. A performance analysis is described in section IV. In section V, we conclude this paper.

## II. Framework of XCCP

We design and prototype an XCCP where conference manipulation information is carried into the XML body to confirm to the concept of the XCON-WG [7].

Each request or response that contains an XML body can be carried on a simple object access protocol (SOAP) [8], configuration access protocol (XCAP) [9], or any other standardized protocol carrying XML. For this paper we chose XCAP for clients, a newly introduced protocol by the IETF SIMPLE WG. XCAP has the advantage of being light, as it binds onto HTTP directly. The conference policy server, however, handles both XCAP and SOAP mechanisms, so that a client running SOAP can also be supported.

Figure 1 illustrates the overall architecture of a designed system, and Fig. 2 presents the protocol stacks of each element. The main architecture is based on the SIP [5] signaling protocol. An H.323-based conference system can communicate with the SIP server using a signaling gateway that supports interworking between the SIP and H.323.

The conference policy server handles the conference control data and the conference application server, which is called focus in a SIP-based conference framework, deals with call signaling data. A media server is connected to the conference application server and sends the media data to the multiple participants.

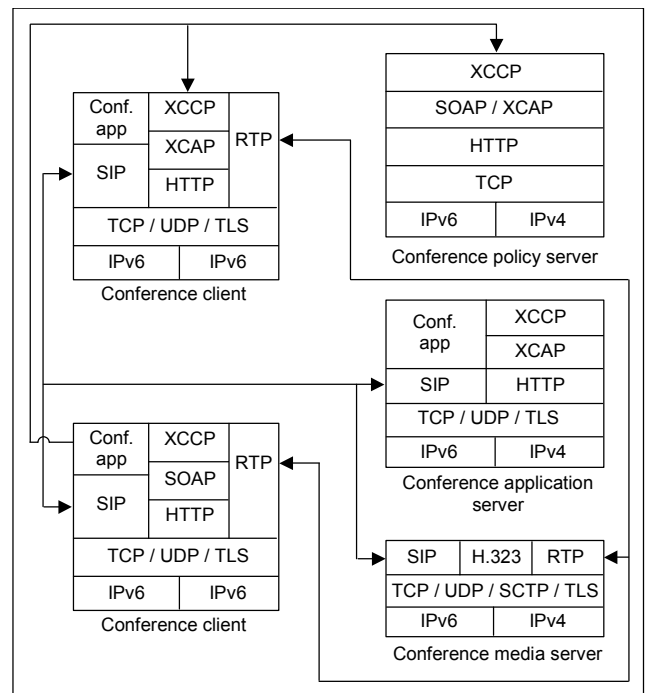


Fig. 2. Protocol stacks in a conference system.

## III. Implementation of Operations

We categorize conference control operations into conference management, membership management, media management and floor management. XML1.0 and Unicode Transformation Format-8 encoding are used for implementation, and we define the conference-control namespace for conference control information. The XML schema is contained in Conference.xsd, and an instance of the schema is named Conference.xml. The schema has the root element, Conference, and sub-elements,

Conf-Mng, Member-Mng, Media-Mng, and Floor-Mng, as shown in the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace=http://conf.etri.re.kr/ conference-control
  xmlns:tns=http://conf.etri.re.kr/conference-control
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="conference">
    <xs:complexType> <xs:sequence>
      <xs:element name="Conf-Mng" type="tns:conf-Type"/>
      <xs:element name="Member-Mng" type="tns:memType"/>
      <xs:element name="Media-Mng" type="tns:mediaType"/>
      <xs:element name="Floor-Mng" type="tns:floorType" minOccurs="0"/>
    </xs:sequence> </xs:complexType> </xs:schema>
```

The operations performed by conference management are creating and destroying conferences, sidebars, or cascading conferences; approving policy changes; and notifying the cascading conference of the changes made. Membership management operations are adding, removing, blocking or ejecting participants; obtaining membership information; and manipulating user privileges. Media management operations are adding and removing media; granting media changes; and changing media policy. Floor management operations are requesting and granting the floor and manipulating the floor controller.

Figure 3 explains an example of a “create conference” operation, while Fig. 4 shows an ‘inviting a user’ operation in an

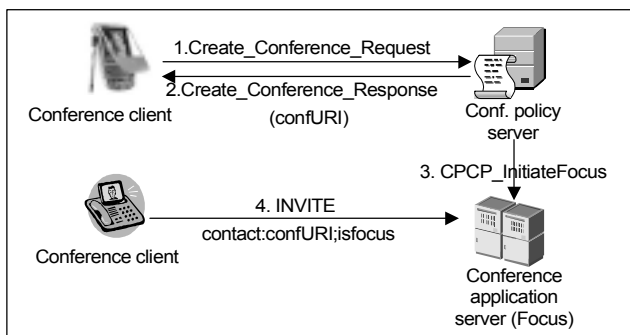


Fig. 3. Create and join operations.

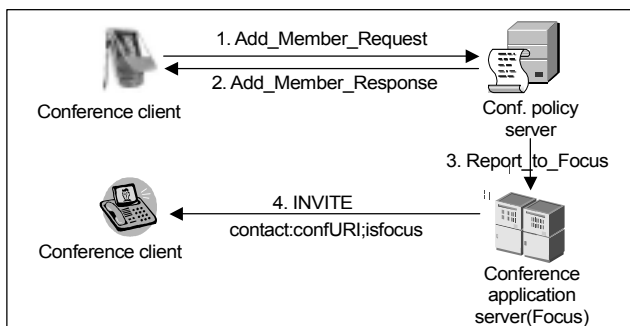


Fig. 4. Invite operation.

XCCP using SIP-based clients.

When the operations are bound to an XCAP, they use HTTP PUT and POST commands as shown in the following:

- Creating a conference:

PUT <http://conf.etri.re.kr/conferences/Conference.xml>? Conference HTTP/1.1  
Content-type: application/conference-control+xml

- Adding a user to a conf.. “sip:conf1@etri.re.kr”:

POST <http://conf.etri.re.kr/conferences/Conference.xml>?  
Conference/Conf-Mng[[@confURI=sip:conf1@etri.re.kr](mailto:confURI=sip:conf1@etri.re.kr)]/ACL HTTP/1.1  
Content-type: application/conference-control+xml

The conference policy server receiving the requests encodes the XML body and creates a conference or adds an access control list of the conference. Receiving the corresponding response from the server, SIP-based clients send a SIP INVITE message to connect with the conference application server. In the case of a change in the access control list, the conference policy server reports the change to the conference application server to initiate an INVITE. If the clients work on H.323, they send a ‘Conference-Join.request’ instead of a SIP INVITE.

## IV. Performance Evaluation

In order to measure the performance of the XCCP, a simulation is performed on a conference policy server using Linux 9.0 and on clients running Windows XP whose requests are carried on an XCAP [9]. The operations of the XCCP are implemented using C language. We calculate the average round trip time, which starts when a client sends a request and ends when the client receives the corresponding response after the conference policy server processes the request and transmits the proper results. Network delay is ignored in this simulation. We run the simulation 1000 times, and record the average value.

First, we simulate the “create conference” operation of XCCP to find out its scalability. Figure 5 illustrates the simulation results. For the number of conferences from 5 to 100, the round trip time for a client’s request within each conference is nearly the same.

Next, we evaluate the scalability of XCCP using the “invite user” operation. Simulation is performed on 1 to 15 conferences

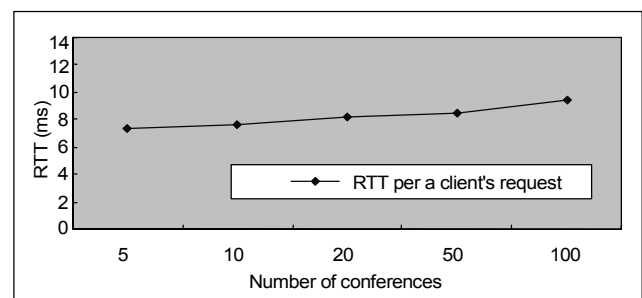


Fig. 5. Simulated round trip time (5 to 100 conferences).

with 5 to 100 users in each.

Figure 6 shows the simulation results and shows that the RTT barely changes until each of the 15 conferences has 20 users, or each of the 3 conferences has 100 users. When each of the 10 conferences has 100 users, and in the cases when both 50 and 100 users are added in each of the 15 total conferences at once, the round trip time increases slightly. From these results, we learn that the total number of users affects the performance of the XCCP rather than affecting the number of conferences. In addition, the XCCP is scalable from small to mid-sized conferences with a total of about 500 users, which we believe is appropriate for centralized conferences.

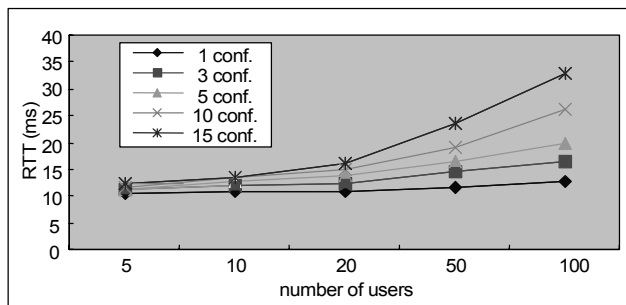


Fig. 6. Simulated round trip time (1 to 15 conferences with 5 to 100 users in each).

## V. Conclusions

This paper describes an XML based conference control protocol. In the XCCP, conference policy data and operations are defined by XML.

The simulation results show that the designed protocol is quite scalable for dealing with an increasing number of conferences. It also shows that the protocol fits for small to mid-sized conferences so it is proper for use in centralized conferences where the current researches on conferencing is being focused.

For future study, we will simulate the designed system under various network environments including network delay and objective complexity of performance.

## References

- [1] N. Kausar and J. Crowcroft, "General Conference Control Protocol," *Conf. Telecommunications*, Edinburgh, 1998, pp. 143-150.
- [2] R. Malpani, L.A Lowe, "Floor Control for Large-Scale Mbone Seminars," *Proc. ACM Multimedia*, Nov. 1997, pp. 155-163.
- [3] ITU-T Rec. H.323, *Packet-Based Multimedia Communications Systems*, Sept. 1999.
- [4] ITU-T Rec. T.120, *Data Protocols for Multimedia Conferencing*, July 1996.
- [5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, *SIP: Session Initiation Protocol*, IETF RFC 3261, June 2002.
- [6] O. Levin, *Conference Policy Control Protocol for Centralized Conferencing*, IETF Internet Draft, June 2003.
- [7] P. Koskelainen, *Requirements for Conference Policy Control Protocol*, IETF Internet Draft, June 2003.
- [8] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer, *Simple Object Access Protocol (SOAP) 1.1*, W3C Note, May 2000.
- [9] J. Rosenberg, *The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)*, IETF Internet Draft, June 2003.