INTERWORKING BETWEEN THE RSW CONTROL CRITERIA AND SIP STANDARD

O. Abouabdalla & R. Sureswaran
School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia

Keywords: Multimedia Conferencing System (MCS), Session Initiation Protocol (SIP), RSW to SIP Protocol (R2SP).

Abstract: Various standards organizations have considered signaling for voice and video over IP from different approaches. There are currently more than one standard for signaling and control of Internet telephone calls. Some of them, which widely used are RSW control protocol and the IETF Session Initiation Protocol (SIP). Both protocols provide comparable functionality using different mechanisms and provide similar quality of service. Although there are numerous industry debates about the merits of the two protocols, the truth is that both of them, along with other complementary protocols, are necessary to provide universal access and to support IP-based enhanced services. Both protocols have been widely deployed, so interworking between RSW and SIP is essential to ensure full end-to-end connectivity. Because of the inherent differences between RSW and SIP, accommodation must be made to allow interworking between the two protocols. The work reported in this paper proposes a communication translation protocol to bridge the RSW control protocol and SIP control protocol. This communication translation protocol has to provide a set of rules to enable communications between the RSW control criteria and SIP standards. The communication translation entity defined can be called translator server.

1 INTRODUCTION

In recent years the use of distributed computer network systems has been increased in many areas of industry, government and academia. The use of computer-based communication applications is becoming more and more important. Video conferencing is hardly a novel concept, with strong interest in the use of video to enhance remote collaboration. Many studies have found that groups are more effective or efficient at solving problems and making decisions when they are connected through a video and audio link compared to when they use only an audio link (Gale, 1990).

The idea of video conferencing debuted in the 1920s (Schooler, 1996). AT&T introduced its PicturePhone at the World Fair in the 1960s and has continued to promise a teleconferencing revolution (Snell, 1994). Clearly, the mission of conferencing and collaborative computing is not only to bring individuals together in space and time, but also to make groups more effective at their work.

Today's multimedia conferencing systems are powerful tools that can revolutionise the way we collaborate in most of our life fields. When people become used to multimedia conferencing for communications with colleagues in their workplaces they will want to include persons from other locations (business or home) in such conferences. As the conferences become larger, the need for flexible presentation control and multimedia combining will become more pressing.

There are many systems used to conduct meetings in a virtual manner to discuss or share ideas, or even to make cheap phone calls. One of the protocols used is the RSW control protocol, which is used in the development of the Multimedia Conferencing System called MCS (Sureswaran and Abouabdalla, 1999; Abouabdalla and Sureswaran, 2000). Another protocol called SIP or Session Initiation Protocol (Handley *et al.*, 1999) is also widely used these days for initiating an interactive user session that involves multimedia elements such as video, voice, chat, gaming, and virtual reality.

2 WHAT IS RSW CONTROL PROTOCOL

The RSW Control Criteria (Sureswaran, 1996; Sureswaran *et al.*, 1997; Sureswaran, 1998) was

Abouabdalla O. and Sureswaran R. (2004).
INTERWORKING BETWEEN THE RSW CONTROL CRITERIA AND SIP STANDARD.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 236-242
DOI: 10.5220/0001384502360242

designed based on how an actual conference is conducted around a meeting table. The RSW Control Criteria is used to address the following two issues with multimedia conferencing:
- The confusion generated when everyone tries to speak at the same time.
- The tremendous amount of network traffic generated by all these participating sites.
It creates a system of order for conferences. The varieties of options proposed within the RSW Control Criteria are:
1- ***Equal Privileges***: all conference sites have an equal opportunity of becoming active sites.
2- ***First come first serve***: assignment of active site status to sites following the order of requests coming in.
3- ***First come first serve, with time-out***: as in option 2 above but with each active site being allowed a certain maximum time limit.
4- ***Organizer Main site***: gives the privilege of choosing the active site to the site that organizes the conference.
5- ***Restricted Active sites***: the organizing site can restrict the sites allowed to participate in the conference.
6- ***Restricted active sites, upgradeable observer sites***: the same as option 5, but with the additional ability to upgrade observer sites to active sites in real-time.

Any one or a combination of these options can be used to control a conference as long as no contradictions arise. In general, a conference is made up of a conference chairman, participants and observers. The conference chairman is the organizer of the conference, while other conference members can be participants or observers.

2.1 RSW history

RSW control protocol has its origins in late 1993 as a control mechanism for multimedia conferencing. It was designed by the Network Research Group in school of computer sciences – University Sciences Malaysia (USM). The first completely working system developed based on RSW was Multimedia Conferencing System MCS version 3.0, which developed on 1996. After that developing MCS version 4.0 on late 1997 extended MCS further. On the beginning of 2003 MCS version 5 was released as beta.

3 WHAT IS SIP CONTROL PROTOCOL

Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences (Malim, 2002).

SIP invitations used to create sessions carry session descriptions that allow participants to agree on a set of compatible media types. SIP makes use of elements called proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers. SIP runs on top of several different transport protocols (Grobel, 2002).

3.1 SIP history

SIP has its origins in late 1996 as a component of the "Mbone" set of utilities and protocols. The Mbone, or multicast backbone, was an experimental multicast network overlaid on top of the public Internet. It was used for distribution of multimedia content, including talks and seminars, broadcasts of space shuttle launches, and IETF (Internet Engineering Task Force) meetings. One of its essential components was a mechanism for inviting users to listen in on an ongoing or future multimedia session on the Internet. Basically - a session initiation protocol. Thus SIP was born (Rosenberg and Shockey, 2000).

As an Mbone tool (and as a product of the IETF), SIP was designed with certain assumptions in mind. First was scalability: Since users could reside anywhere on the Internet, the protocol needed to work wide-area from day one. Users could be invited to lots of sessions, so the protocol needed to scale in both directions. A second assumption was component reuse: Rather than inventing new protocol tools, those already developed within the IETF would be used. That included things like MIME, URLs, and SDP (already used for other protocols, such as SAP). This resulted in a protocol that integrated well with other IP applications (such as web and e-mail).

Despite its historical strengths, SIP saw relatively slow progress throughout 1996 and 1997. That's about when interest in Internet telephony began to take off. People began to see SIP as a technology that would also work for VoIP, not just Mbone sessions. The result was an intensified effort towards completing the specification in late 1998, and completion by the end of the year. It received official approval as an RFC (Request for Comments, the official term for an IETF standard) in February 1999, and issuance of an RFC number, 2543, in March.

From there, industry acceptance of SIP grew exponentially. Its scalability, extensibility, and - most important - flexibility appealed to service providers and vendors who had needs that a vertically integrated protocol, such as H.323, could not address. Throughout 1999 and into 2000, it saw adoption by most major vendors, and announcements of networks by service providers. On June 2002 IETF published SIP most recent version, RFC 3261 (Morrissey, 2003).

4 PROPOSED COMMUNICATION TRANSLATION PROTOCOL (R2SP)

It is based on the two control protocols discussed above that we propose a communication translation protocol to bridge the RSW control protocol and SIP control protocol**.** This communication protocol will perform the functions of a 'translator' so that any MCS client user can communicate with SIP client (SIP soft phone or SIP IP phone) user. Since MCS is build based on RSW, we used it as the base for our translation protocol.

4.1 Interworking between RSW and SIP

Interworking between RSW and SIP is based on MCS version 4.0 and SIP version 2.0. The goal of interworking between RSW and SIP requires transparent support of signaling and session description between the SIP and MCS entities. The server provides this translation of RSW-SIP is called the translation server (R2SP). The translation server (R2SP) that will allow interworking between the MCS and SIP network can be architected in a variety of ways, which are co-existence with SIP server, or without SIP server. In this paper we discuss the system without existence of SIP server.

Interworking between MCS and SIP may involve in the following entities:

- **MCS Server:** The MCS server is an entity on the network that performs the functions of a controller to a conference. It provides users a platform to register/login to participate in conferences. It also provides other services such as multicast address assignments and providing damage control when links break.
- **MCS Client:** A MCS client is an endpoint on the network, which provides the real-time, two-way or multiple way communications with another MCS clients. This communication consists of control, indications, audio, video and/or data between the MCS clients.
- **Translation Server (R2SP):** It allows interworking between MCS and SIP networks. The MCS side of the R2SP is the part of the R2SP that terminates and originates MCS signaling from and to the MCS network respectively. The SIP side of the R2SP is the part of the R2SP that terminates and originates SIP signaling from and to the SIP network respectively.
- **SIP User Agent (UA):** A logical entity that can act as both SIP user agent client, and SIP user agent server.

The R2SP supports the address resolution schemes of both MCS and SIP. It registers itself to the MCS server. When the R2SP receives signaling messages from MCS server, it performs the necessary translation and sends the corresponding equivalent messages to the SIP entity on the SIP side of the R2SP and vice versa. The R2SP provides signaling translation for all phases of a call or a conference. The R2SP has a table of reference for lookup to resolve MCS and SIP addresses. Fig. 1 shows the internetworking configuration of the system.

R2SP may contain the functions like Call sequence mapping, Address resolution, Terminal Capability transaction, Opening and closing of media channels, Mapping media algorithms for MCS and SIP network, Call resource reservation and release, Ability to provide the state of a call, Call state machine, Mid Call signal processing, Service Interoperability Logic, and media processing.

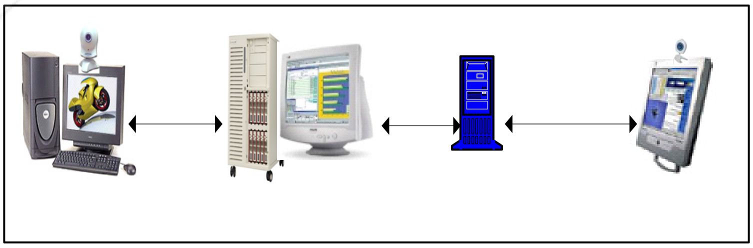

Figure 1: Interworking between MCS and SIP

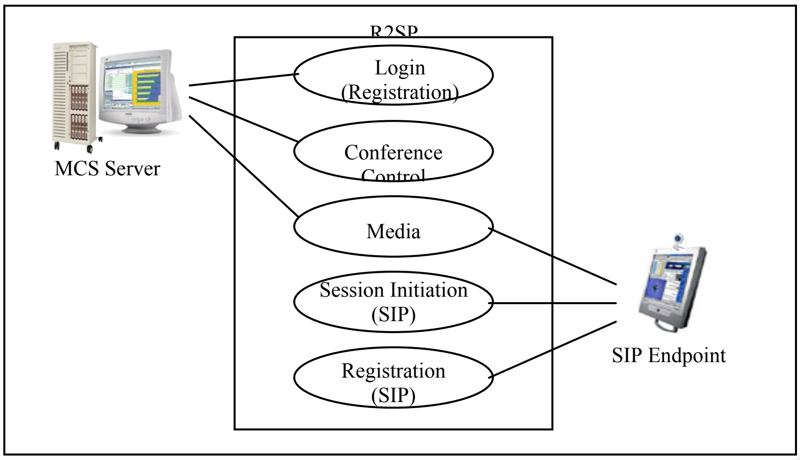

Figure 2: R2SP Use Case Diagram

R2SP maintains conference message sequence on both sides in such a way that neither MCS client nor SIP UA is aware of the R2SP presence. The R2SP provides seamless interworking between the conference flows of the two protocols. The messages that do not have match on the other side should be terminated on the R2SP, and R2SP takes the necessary action on them. The messages and parameters, which do not have direct mapping on the other side, are to be generated by the R2SP with default parameters in most cases. The R2SP conforms to the conference signaling procedures recommended for the MCS side independent of the SIP side. Also, the R2SP conforms to the call signaling procedures recommended for the SIP side independent of the MCS side.

4.2 System Module

In MCS, there are two types of registration. MCS server should register it-self to other MCS servers (since R2SP works as MCS server it should register to other MCS servers), the second type of registration is the process by which an MCS client login to MCS server, and informs it of its IP address. Registration will occur before any conferences are attempted. The MCS server will respond with either a confirmation or a reject message. In SIP, the REGISTER request allows a client to let a proxy or redirect server know its current address. In this case, the R2SP will have the look-up tables for SIP address resolution, since SIP users register to it.

In general, the R2SP will contain the functions needed to establish and tear-down the conference such as: opening and closing of media channels, mapping media algorithms for MCS and SIP network, call resource reservation and release, ability to provide the state of a call or conference information and mid call or conference signal processing. Media processing within the R2SP will be minimal. It is assumed that the same transport protocols (e.g., RTP, TCP, UDP, etc.) will be used in both MCS and SIP networks for carrying media.

Interworking between MCS and SIP may involve in two types of Endpoints: MCS clients and SIP User Agents (UA). Other entities may include MCS-SIP translation server (R2SP) and MCS server.

4.3 System Components Analysis

We analyze the function modules of each one of the system components. Fig. 2 shows use case diagram for R2SP. Since R2SP is an entity of interworking function for message translation between MCS and SIP, it should include all necessary modules in MCS server and SIP EP. R2SP should also contain the message-mapping module for call signaling translation, and keep the state of call setup.

R2SP has the login (registration) admission control module to serve for registration from other MCS servers. The R2SP contains the module for forwarding conference setup and control messages. It also contains the module for forwarding media. For SIP side, R2SP contains registration module for SIP EP registration and address resolution, and session initiation module to forward session initiation messages.

4.4 Call setup translation

Three pieces of information are needed for establishing a call between two end points, namely the signaling destination address, local and remote media capabilities, and local and remote media transport addresses at which the endpoint can receive the media packets. In MCS, this information is spread over different stages of the call setup, while SIP conveys it in an INVITE message and its response. Translating a SIP call to MCS call is straightforward. The R2SP gets all the information in the SIP INVITE message and split it across multiple stages of the MCS call establishment. However, in the reverse direction, from MCS to SIP, the different stages of MCS call establishment have to be merged into a single SIP INVITE message.

When SIP EP initiates a call it send an INVITE message to R2SP which includes the address of the invited person. This address is in a form xyz@domain. R2SP when receive the message it split it into two parts, user name and domain name, search for the domain name in its server table and then send NOTIFY indication which contains invited user name to desired MCS server.

When MCS client want to start a conference it will send USER_LIST request to its MCS server, which forward it to R2SP. R2SP send back the user list, MCS client then send create message to its MCS server, which then send NOTIFY indication to R2SP. R2SP once receive the NOTIFY indication it will combine the information and send INVITE request to desired SIP EP.

R2SP must also map session descriptions between the two signaling protocols. MCS Version 4.0 uses a wavelet codec by default for video, and it uses the 8 bit 11kHz PCM for audio. SIP can, in principle, use any session description format. In practice, however, SDP (Handley and Jacobson, 1998) is used exclusively. SDP lists media types and the supported encodings for each. Thus, a MCS media capability can be easily described in SIP. R2SP will use SDP to send media capability to SIP EP and includes audio/video codecs and port number in the SDP.

4.5 Connection establishment and call ending

Once the user knows that the destination is reachable via the R2SP, the connection is established. A point-to-point call from Ali to Sara needs three crucial pieces of information, namely the logical destination address (A) of Sara, the media transport address (T) at which each of the users is ready to receive media packets (RTP/RTCP) and a description of the media capabilities (M) of the parties. Ali should know A, T and M of Sara while Sara needs to know Ali's T and M. The difficulty in translating between SIP and MCS arises because A, M, and T are all contained in the SIP INVITE request and its response, while MCS may spread this information among several messages.

R2SP uses A, M and T for establishing a conference with MCS. The responses from the MCS side are collated and forwarded to the SIP side. The R2SP may get the media capabilities of the SIP user agent using the SIP OPTIONS message. Media capabilities of the MCS terminal are fixed. Once the logical channels are established from the R2SP to the MCS server, the R2SP knows M and T and can place a SIP call by sending an INVITE.

Fig. 3 shows a session setup example. In this example Ali from SIP client want to establish a call with Sara from MCS network. First Ali has to send REGISTER request to SIP registrar at R2SP (F1), R2SP (*r2sp.usm.my*) once receive it, it will add new record to its users table with Ali's information and then send back 200 OK response to Ali's SIP client (F2). If a registration could not be done, an error response would have been sent instead of the 200 OK. On the other hand Sara should register to its MCS server by sending LOGIN message to *mcs.nrg.usm.my* server (F1a) and wait for the LOGIN_OK response message from the server (F2a).

Ali after registration sends INVITE request to *r2sp.usm.my* (F3). The INVITE request contains a number of header fields. The ones present in an INVITE include a unique identifier for the call, the destination address, Ali's address, and information about the type of session that Ali wishes to establish with Sara. When *r2sp.usm.my* receive this request, it looks through its servers table to find *mcs.nrg.usm.my* information then sends NOTIFY message to it, with Sara as an invited user (F4), which then forward the NOTIFY message to Sara's MCS client (F6). At the same time *r2sp.usm.my* sends 100 Trying to Ali's SIP client (F5), to indicate that the INVITE has been received and that R2SP is working on his behalf to route the INVITE to the destination.

When Sara receive the NOTIFY message, and find out that he invited to a conference, he will send JOIN message to *mcs.nrg.usm.my* server (F7), which then forward the JOIN message to *r2sp.usm.my* (F8). When *r2sp.usm.my* receive the JOIN message, it will send 180 Ringing response to Ali's SIP client (F9). At the same time Sara's MCS client will send REQ-ACTIVE message to *mcs.nrg.usm.my* server (F10), which then forward the message to *r2sp.usm.my* (F11). When *r2sp.usm.my* receive the REQ-ACTIVE message, it will send 200 OK response to Ali's SIP client (F12), which indicate that Sara agree to call establishment and ready for media transaction.

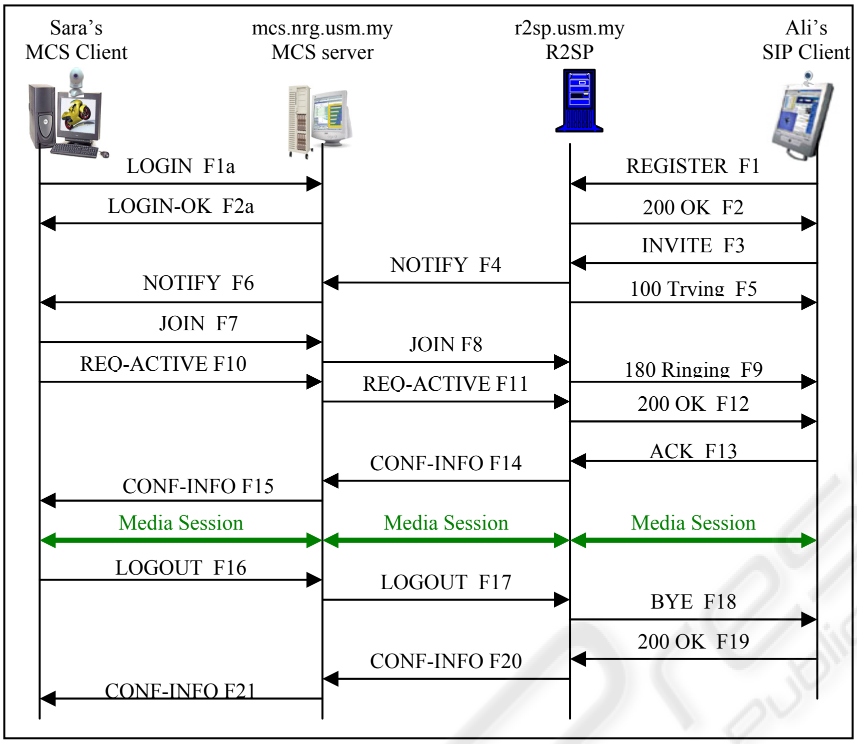

Figure 3: SIP to MCS session setup example

At this point Ali's SIP client will send acknowledgement message ACK to *r2sp.usm.my* (F13), which when receive the ACK request, sends

Ali and Sara's media session has now begun, and they send media packets using the format to which they agreed in the exchange of SDP between R2SP and Ali's SIP client. They usually use MCS version 4.0 UDP 7600 port as default for audio and UDP 7601 port as default for video. At the end of the call (conference), Sara disconnects (hangs up) first and generates a LOGOUT message to its MCS server (F16), which then forward the message to *r2sp.usm.my* (F17). When *r2sp.usm.my* receive the LOGOUT message it generates BYE message to Ali's SIP client (F18). Ali confirms receipt of the BYE with a 200 OK response (F19), which terminates the session in the SIP side. To terminate the session in the MCS side, *r2sp.usm.my* sends CONF_INFO message to *mcs.nrg.usm.my* server (F20) which contains conference end indication and free all call (conference) resources. The *mcs.nrg.usm.my* server, then forward the CONF_INFO message to Sara's MCS client (F21).

5 CONCLUSION

The work in this paper has thus provided modeled and verified translation protocol of interworking

CONF_INFO message to *mcs.nrg.usm.my* server (F14). The *mcs.nrg.usm.my* server, then forward the CONF_INFO message to Sara's MCS client (F15).

between MCS and SIP. From our work, we conclude that the fact of our design for the system of interworking between MCS and SIP is good by using SDL, which is proved as a simple and very efficient method to verify and validate protocols.

We have verified most of the successful scenarios and some of the failure scenarios. Besides, we use ObjectGeode to help the verification of dynamic behavior of our system model. Finally, we identified that the advanced feature of the communication translation protocol and advanced service based on MCS-SIP system is a hot research topic and being currently investigated.

Further research work to enhance the translation protocol is at least in the following two areas:

(a) Since our design is based on MCS version 4.0, and MCS version 5.0 is out, R2SP can be expanded to support the new version of MCS as well as support any new SIP version.

(b) Integrate the different models into a single, but much larger system. For example we can integrate H.323 or MGCP models and MCS-SIP Interworking system model into larger system to find more useful property.

Finally, our model can be also a starting point to begin the research of 3GPP network since SIP has been defined as signaling protocol in next generation network architecture.

REFERENCES

Abouabdalla, O. and Sureswaran, R. (2000). A Server Algorithm to Manage Distributed Network Entities for Multimedia Conferencing System. In Proceedings of IWS (Internet Workshop on Asia Pacific Advanced Network and its Applications). Tsukuba, Japan. Feb 2000. pp. 141-146.

Gale S. (1990). Human aspects of interactive multimedia communication. Interaction with Computers.

Grobel, I. (2002). SIP is a key part in multimedia sessions. Network World**.** Framingham. Aug 2002, Vol. 19, Iss. 32, pp. 35.

Handley, M. and Jacobson, V. (1998). SDP: session description protocol. Request for Comments (Proposed Standard) 2327. Internet Engineering Task Force. Apr 1998.

Handley, M., Schulzrinne, H., Schooler, E. and Rosenberg, J. (1999). SIP: Session Initiation Protocol. Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, March 1999.

Malim, G. (2002). Opportunity Knocks. Communications International. London. Jul 2002, pp. 16.

Morrissey, P. (2003). It's time to take a look at SIP. Network Computing. Manhasset. Apr 2003, Vol. 14, Iss. 7, pp. 76-79.

Rosenberg, J.D. and Shockey, R. (2000). The Session Initiation Protocol (SIP): A Key Component for Internet Telephony. Computer Telephony. June 2000.

Schooler, E.M. (1996). Conferencing and collaborative computing. Multimedia Systems. Vol. 4, pp. 210-225.

Snell, M. (1994). Picture this: videoconferencing. IEEE Computer. Vol. 27, pp. 8-10.

Sureswaran, R. (1996). A Reflector Based System to Support the RSW Multimedia Conferencing Control Criteria. IASTED International Conference on Networks, Orlando, January 1996.

Sureswaran, R., Subramanian, R.K., Guyennet, H. and Trehel, M. (1997). Using the RSW Control Criteria To Create A Distributed Environment for Multimedia Conferencing. In Proceedings of REDECs ’97. Penang, Malaysia. 27-29 November 1997.

Sureswaran, R. (1998). A Distributed Architecture to support Multimedia Applications Over the Internet and Corporate Intranets. In Proceedings of SEACOMM ’98. Penang, Malaysia. 12-14 August 1998.

Sureswaran, R. and Aboudallah, O. (1999). A Server Recovery Procedures to Manage Distributed Network Entities for Multimedia Conferencing System. In Proceeding of World Engineering Congress (WEC99), University Putra Malaysia, Kuala Lumpur. July 1999. pp. 81-85.