Different Approaches of interworking between SIP and H.323

Shifta Ansari[1], Kashif Nizam Khan[2], Jinat Rehana[2], Nusrat Jahan Lisa[1], Shahriar Kaisar[3]

1. Department of Computer Science and Engineering Ahsanullah University of Science and Technology(AUST)
2. Helsinki University of Technology
3. Dept. of Computer Science & Information Technology, Islamic University of Technology (IUT)

Abstract
Currently two standards exist for signaling and control of voice over IP calls, namely ITU-T Recommendation H.323 and the IETF Session Initiation Protocol (SIP). Although there are a significant number of similarities between these two protocols, they behave significantly different when providing different higher level services. On one hand, SIP provides flexibility with broader scope, offering functions specifically designed to enable easy extensions of new technologies. On the other hand, H.323 stands still with more established standard providing better framework and interworking. This paper describes some noteworthy differences between them and presents some solutions to make these two giant players co-existing.
***Keywords**: VoIP, SIP, H.323, Gateway, RTP, RCTP, H.225, H.450, SDP*

1. Introduction

Multimedia Communications over Internet Protocol (IPMC) has opened a door to almost endless possibilities for us especially in terms of communication capabilities for end users. Unlike in years past, when we had to hang over a "cring cring" telephone and wait for hours only to make a long distance voice call, the Internet telephony has given us the greatest platform for communication. Internet telephony integrates a variety of services provided by the Internet and PSTN (public switched telephone network) infrastructure. These services include voice chatting, videoconferencing, web collaboration, instant messaging services, white-boarding, application sharing, and other forms of multimedia communication offered from a wide number of service providers [6]. Communication over internet protocol (IP), specially the Voice over IP (VoIP) has revolutionized the telecommunications industry. In one hand, it is providing voice calls with lower call fees and simplification of deployment and on the other hand, it is providing a common platform for voice and data networks. Moreover, the deployment of such communication system is simpler than other systems, and it also ensures greater integration with multiple applications that offer enhanced multimedia functionality. At this point we must note that, VoIP concerns mainly on the transport of traditional voice phone calls over an IP network. On the other hand, videoconferencing is more or less concentrated on providing communication using voice and video capabilities. But, IPMC is a broader concept which incorporates all forms of end-to-end communication capabilities as we mentioned above. Unlike the Public Service Telephone Network system (PSTN) system, which requires two independently built and maintained systems: one for voice traffic and the other one for call signaling, VoIP (or IPMC in a broad sense) uses a common infrastructure namely the Internet. In order to provide useful services, Internet telephony requires a set of control protocols for the purposes of connection establishment, capabilities exchange, and conference control. For signaling, VoIP has a sweet choice confusion between two families of very effective protocols: H.323 and SIP. Both the Session Initiation Protocol (in conjunction with Session Description Protocol (SDP)) [9] and the H.323 [8] are primarily used for setting up internet multimedia conferences and telephone calls. For example, currently H.323 is the most widely used protocol for PC-based conferences, due to the widespread availability of Microsoft’s NetMeeting tool, while carrier networks using so-called soft switches and IP telephones are based on SIP [14]. In order to achieve universal connectivity, interworking between these two protocols is desirable. In this text, we have mainly focused on two areas: firstly, we have presented a comparative study of these two application layer protocols and secondly we would like to introduce some possible interoperability options between them.

2. Protocol Overview

In this section, we will discuss an overview of how these two protocol bodies play on a real time network. Let’s start with SIP.

2.1 SIP

Session Initiation Protocol (SIP) is a standard introduced by the Internet Engineering Task Force (IETF)[4, 5, 3]. It was introduced in 1999 and its main focus was to carry voice over IP. The main functionality of SIP includes signaling and session management within a packet

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

telephony network. Through signaling, SIP carries the call information across network boundaries and session management provides the ability to negotiate the attributes of an end-to-end call. SIP can provide many types of functionalities. Its capabilities include:

- Location determination of the end point: One of the major functionalities of SIP is providing address resolution. It also supports name mapping, and call redirection.

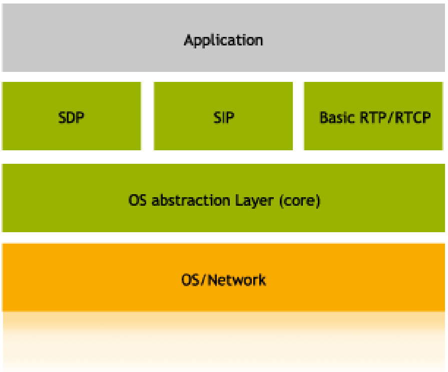

Figure 1: SIP Protocol Stack [13]

- Negotiation of media capabilities of the end points: To determine the common services and negotiate the capabilities between end points, SIP uses Session Description Protocol (SDP). In case of conferences, SIP ensures that all the end points do agree on a common set of capabilities.
- Determining the availability of the end point: SIP can determine the reason for unavailability of the target end point, just as the mobile networks do. For example, SIP can determine whether the called party is already on the phone or did not answer during the allotted number of rings. It then returns a message indicating why the target end point did not answer.
- Session Establishment: SIP is mainly designed to establish, maintain and tear down sessions between endpoints. It also supports call changes at the middle of a running conversation, such as the addition of a new end point to a conference.

- Handling call transfer and call termination: One of the dynamic functionalities of SIP is the support of call transfer from one end point to another. When a call transfer is initiated, SIP simply establishes a session between the transferee and a new end point and terminates the previous session between the transferee and the transferring party.

2.2 H.323

H.323 works a little bit differently in comparison to SIP. H.323 is an umbrella specification, meaning that it is not a protocol by itself, but rather defines how to use other protocols. H.323 was developed by the International Telecommunications Union (ITU) in 1996.From Fig. 2 shows that H:323 standard consists of several protocols, including above all H.225 RAS signaling, H.225.0 Call signaling(Q.931), H.245 Control signaling, RTP,

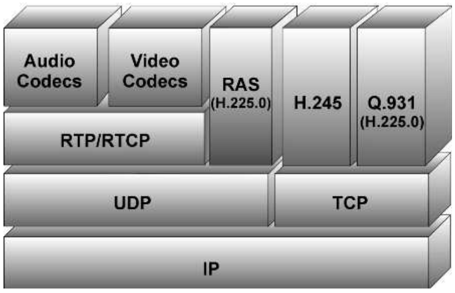

Figure 2: H.323 Protocol Stack [2]

RTCP, H450 Supplementary services and other standards for voice and video digitization and compression [7, 8]. The H.323 standard provides the system and component descriptions, call model description, call signaling procedure, control messages, multiplexing, audio and video codec, as well as the data protocols for the developers. For call controlling and signaling it uses H.225.0, H.225.0/RAS and H.245. H.225.0 is mainly a call signaling protocol. It uses a subset of Q.931 signaling protocol for media stream packetization. H.225.0/RAS functions for user registration, admission and status signaling. H.245 is responsible for controlling of multimedia communications. For audio and video processing H.323 incorporates several protocols. For audio processing it uses G.711, G.722, and G.728 and G.729. These protocols provide modulation and speech coding at different speeds. H.261 and H.263 protocols are responsible for video processing providing different video codec for audiovisual services. H.323 also bundles the protocol suite T.120 for data transmission between end points. It can be used for various applications in the field of collaboration Work, such as white boarding, application sharing, and joint document management. The media transportation protocols for H.323 are same as SIP namely RTP/RTCP. H.235 provides security and encryption for H series multimedia terminals. H.323 also incorporates the protocols H.450.1-H.450.12 to provide some supplementary services like call transfer call diversion, call hold and so on.

2.3 Call Establishment

The working principle of SIP is quite simple. Basically, it has two elements: the SIP user agent (containing User Agent Server, UAS and User Agent Client, UAC) and servers (which can assume various roles, like, redirect, proxy, and registrar). Fig. 3 illustrates the establishment of SIP session between two user agents (UAs). Both the UAs register their availability and their IP addresses with the SIP registrar in their own ISP's network. At the beginning, the call initiator informs its own proxy server that he wants to call or contact) the other UA. The proxy server forwards this request to the callee's proxy server (with the help of a SIP redirect server). When the request reaches the callee, a signal, like a ringtone for example, is send to the call initiator through the proxy servers. At this very point, both the parties know each other's IP and so they can now begin the real time multimedia communication through direct channels (Point-to-point). SIP uses RTP (Real Time Protocol) or RTCP (Real Time Control Protocol) for real-time data transmission.

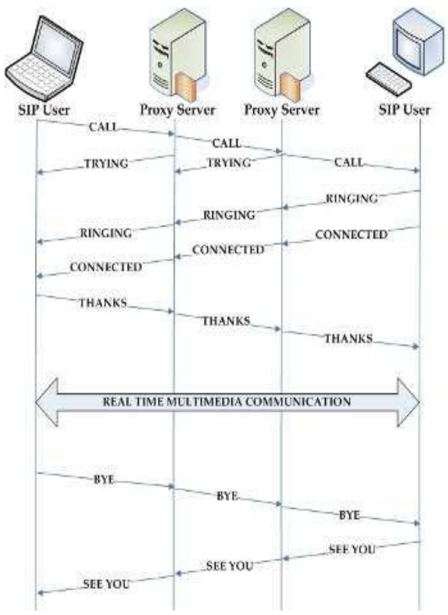

Figure 3: SIP Call Establishment

On the other hand, H.323 works in conjunction with four basic elements namely: Terminals, gatekeepers, Gateways and Multiconference Unit (MCU). Fig. 4 shows the call establishment procedure of two H.323 users. As we can see, lots of messages are communicated between the communication parties and with the gatekeeper also. The call initiator needs to register itself with the gatekeeper before establishing a call request.

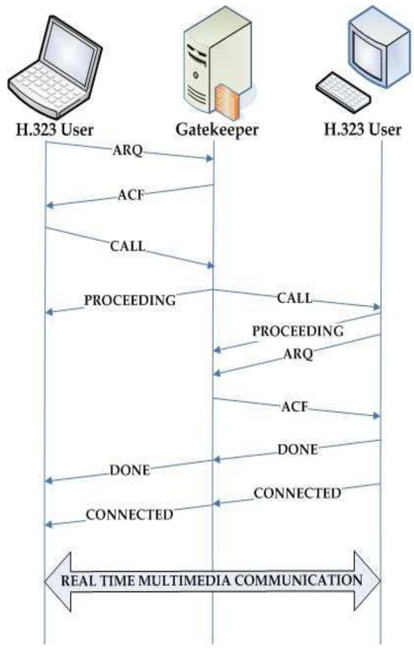

Figure 4: H.323 Call Establishment

The gatekeeper then, tries to connect to the callee. The callee also needs to register itself with the gatekeeper before the connection is established. After, negotiating all the parameters, the parties start communicating each other over a direct channel through the help of RTP or RCTP like protocols.

3. Similarities and Differences

As the motivation behind the protocol suits SIP and H.323 were somewhat similar, a good amount of similarity exists between them. But, at the same time they do differ in various aspects either in architectural issues or in the services they provide. Let's look at some of their similarities and differences in brief. The motivations behind the design of H.323 and SIP were mainly focused on providing services across IP networks. That's why we find an ample amount of similarities between them such as: they both run over IP, and both of them use TCP and UDP sessions for signaling and use Real Time Protocol (RTP) for transmitting the voice and/or video stream over the network. Also, if we look at Sec. 2.1 and Sec. 2.2 we will find that neither of these two protocol families have introduced any new coding or decoding methods of their own (for audio and video transmission), but instead, they have leveraged other existing protocols. Sec. 2.3 shows us that, SIP and H.323 both typically use a server to act as a middleman for setting up calls. In case of H.323, it is a gatekeeper which sends and receives different signaling packets to terminals so that they can set up their media streams to PSTN gateways. On the other hand, if we now look at SIP, it uses a proxy server to process and forward requests from user agents to set up calls directly to other user agents, or through gateways. If we now look at the other side of the coin, H.323 was the first protocol suite to be specially developed for VoIP in 1996 and now it's in its fifth version. The sixth version being on the design table, H.323 was basically based on telephony protocols like ISDN Q.931. On the other hand, IETF design heads first revealed SIP in the mid-90's and have published two RFCs since then, the latest (RFC 3261) in 2002. SIP is basically based on text-based protocols like HTTP and SMTP. These protocols are programmer friendly as these are well understood by many programmers. This is simply because SIP is fairly simple to code and troubleshoot whereas H.323 is written in binary code. It makes it harder to understand for a programmer and requires significant experience and the development tools. One of the major differences between the protocols is that SIP is simply used to set up and tear down media sessions. So, it gives the luxury of choosing the media services for the session, but, H.323 specifies in detail which underlying protocols will be used to provide a specific media service. With SIP, the media itself is independent of the signaling protocol. In fact, as we have mentioned earlier, SIP relies on another protocol SDP to negotiate the media capabilities. So we can easily observe that SIP being the more flexible one, is usable in many areas and it attracts the developers to get more creative with SIP. But by saying that we also should keep in mind that H.323, being the older one, can still be a better choice for the vendors in many cases [1].

4. Interworking between SIP and H.323

The coexistence of two protocols that are incompatible with each other is a real problem for users. It is because the users have to decide and choose between two solutions that have both advantages and disadvantages. If we analyze the market trends, we will find that most of the commercial products use H.323, but they are only supported on a limited number of platforms, do not use IP multicast, and require the use of expensive servers for multi-point conferencing [11]. SIP is used by the MBone tools [10], which are freely available on a number of platforms and use IP multicast for multi-point conferencing, but they are not as well supported or as user friendly as commercial H.323-based systems. Now this situation is motivating the researchers to explore the interoperability between the two standards. Interworking between SIP and H.323 requires transparent support of signaling and session descriptions between the SIP and H.323 entities. Thus some sort of signaling gateway is needed to provide signaling translations between them. In [12], the authors call the server providing this translation a SIP-H.323 interworking function (IWF). In this paper, we will designate it as SHT (SIPH. 323 Translator). We will now present some of the interoperability requirements [12]:

4.1 Protocol compliance

The SHT should have the convenience of using the components of both the H.323 and SIP. So, it should provide and handle all mandatory features of H.323 as well as those of SIP. Common call scenarios should be simple to implement.

4.2 User registration

The users should be free to dial any address, irrespective of knowing that whether it belongs to SIP or H.323. The SHT should use the user registration in both the H.323 and SIP networks to resolve the user name (alias or URL) to an IP address.

4.3 Message Mapping

The SHT is expected to map all the mandatory H.245 messages to appropriate SDP messages and vice-versa, without the endpoint being aware that such conversion is taking place.

4.4 Efficient data communication between the endpoints

Where possible, the SHT should route RTP and RTCP traffic directly between the endpoints involved in the conference without going through the SHT. This reduces the delay for media packets and helps building scalable SHTs.

4.5 Call sequence mapping

The SHT should map the message sequence between H.323 and SIP in such a way that every important decision (accepts or reject a call, choose an algorithm for a logical channel, and so on) is taken by the endpoints involved in the conference and not by the SHT itself.

5. Proposed Solutions

In this section, we will discuss about the solutions of two major interworking problems between SIP and H.323, namely the User Registration problem and the Message Mapping problem. We have tried here to analyze one of the better solutions provided so far for these problems which have been presented in [12].

5.1 User Registration Translation

The User registration servers are the entities which store user registration information. In case of SIP, the user registration server is SIP registrars and it is gatekeepers for H.323. A big challenge for interworking between SIP and H.323 is, to locate users residing in different networks. A H.323 gatekeeper has to know the location of a SIP entity when it is being called by a H.323 end user. But, the question is "how?” A gatekeeper can locate a H.323 entity but it cannot locate a SIP entity and vice versa. Here enters the SHT and it simplifies locating users independent of the signaling protocol. The SHT demands the direct access to user registration servers to provide this simplicity. The user registration server can then forward the registration information from one network, to which it belongs, to the other. We will discuss three different approaches here. Let’s start with the first one:

5.1.1 SHT Residing With SIP proxy/Registrar

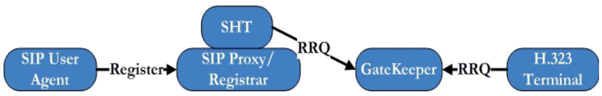

Figure 5: SHT Residing With SIP Proxy/Registrar

In this approach, the SHT works in conjunction with a SIP registrar and proxy server, as shown in Fig. 5. The H.323 gatekeepers here maintain all the registration information. This approach works in the following manner:

1. SIP registrar receives a REGISTER request from an UA.
2. The SHT, residing with the registrar, generates a registration request (RRQ) to the H.323 gatekeeper.
3. Now, the H.323 gatekeeper knows the registration information of the SIP UA.
4. An H.323 user registers usually H.225.0 procedure.

So, now calls can be forwarded in any direction namely from H.323 to SIP or SIP to H.323.

Suppose, if a SIP user agent wants to talk to another H.323 user the scenario will be something like this:

1. SIP UA sends an INVITE message to SIP server
2. The SHT sends location requests (LRQ) to the multiple H.323 gatekeepers (using multicast).
3. The gatekeeper who holds the expected registration information responds with the IP address of the H.323 user.

4. The SIP server now knows the IP address of the callee and so it can now route the call to the destination.

5.1.2 SHT Residing With H.323 Gatekeeper

Figure 6: SHT Residing With H.323 Gatekeeper

This approach is more and less similar to the previous architecture. The only difference here is that, the SHT is collocated with the H.323 gatekeeper as shown in Fig. 6. It works in the following manner:

1. H.323 gatekeeper receives the RRQ request.
2. The SHT forwards a REGISTER request to the SIP registrar.
3. The SIP registrar now knows the registration information of the H.323 UA.
4. H.323 terminals will now appear as SIP URLs to a SIP UA.

Now, any call can now be made between the H.323 and SIP network. If a H.323 entity wants to talk to a user who happens to reside in the SIP network, the call scenario will be something like this:

1. H.323 UA sends an Admission Request (ARQ) to its gatekeeper.
2. The gatekeeper sends an LRQ request.
3. The SHT grabs this request and finds the address in the SIP network by sending a SIP OPTIONS request.
4. If the SHT finds the address in the SIP network it responds with a Location Confirmation (LCF) request.
5. The gatekeeper now knows the IP address of the callee and so it can now route the call to the destination.

5.1.3 Independent SHT

Figure 7: Independent SHT

In the last approach, as shown in Fig. 7, the SHT is not collocated with either with H.323 gatekeeper or SIP proxy server. User registration is done independently in the SIP and H.323 networks. We will now show the address resolution mechanism. If a SIP user calls a H.323, the SHT will function in the following way:

1. H.323 user is already registered with the gatekeeper using RRQ.
2. SIP UA sends a call request for the H.323 user, to its proxy.
3. SIP proxy contacts with SHT for the address resolution.
4. SHT sends a multicast LRQ request to all gatekeepers
5. If the SHT receives a LCF within a certain period of time it sends the address back to the SIP proxy, otherwise confirms it that the address does not exists.

So, this is how SHT works in the SIP to H.323 direction, if we now look at the other possible situation when a H.323 user calls a SIP UA, the SHT works as follows:
1. SIP UA is already registered with the registrar using REGISTER.
2. H.323 user sends a call request to the gatekeeper.
3. H.323 sends an LRQ to the SHT.
4. SHT sends a multicast SIP OPTIONS request to the SIP location servers.
5. On receiving the location confirmation by the location servers, SHT sends the address back to the gatekeeper.

5.2 Message Mapping

Once a user knows the destination address, it can reach the user at the other end. But, the next problem arises when the end users want to negotiate the options for the current session. At a typical point-to-point call scenario, the communicating parties need to know three pieces of important information; self and remote signaling address, self and remote media capabilities and self and remote media transport Address.SIP combines these three pieces of Information in its INVITE request but H.323 spreads them into different stages. So, the problem is to map the multistage signals of H.323 to a single stage INVITE signal of SIP. Although, H.323V2 has a single stage Fast-Start, it is optional and so we will discuss only the signal mappings without Fast-Start. The solution presented for message mapping in [12] is straightforward and not so difficult to implement. The SHT obtains the three pieces of

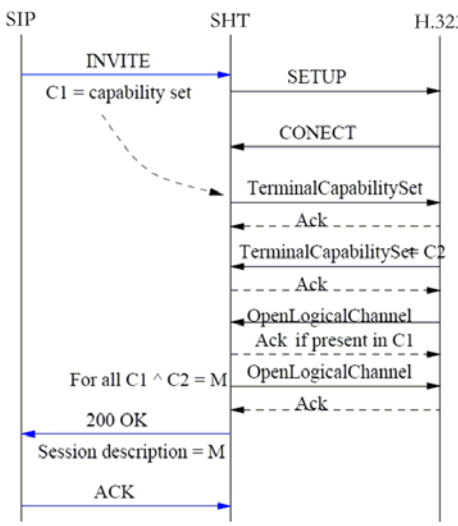

Figure 8: Call from SIP terminal to H.323 terminal [12]

information from H.323 user using Q.931 and H.245 phases. It can also obtain this information from SIP user using INVITE and OPTIONS. It functions in the following way:
1. A SIP UA is calling a H.323 terminal.
2. SHT sits between them and accepts the call on H.323 terminal's behalf.
3. SHT gets the media capabilities of SIP in its INVITE request.
4. SHT obtains the media capabilities of H.323 via H.245 capability negotiation.
5. SHT performs the option negotiations.
6. Upon receiving the OpenLogicalChannel acknowledgement SHT sends a 200 OK to the SIP UA.
7. The logical channels are established and end-to-end media communication can now proceed without the help of SHT as both parties know each other's address now.

Fig. 8 describes the above scenario. Although the mapping looks quite simple, there can be some problems in the real time environment especially from the H.323 terminal's side. This is simply because there are differences in the session description protocol functionalities in both sides. It restricts the change of media options (algorithms for example) while a call is on process. If a parameter changes, SHT might come into play, but it will result in poor scaling. If we now look the other way around, a H.323 terminal calling a SIP UA, the signal translation becomes more interesting and yes, a bit complicated. Fig. 9 illustrates this situation. Let's look at this in more detail:

1. As we already mentioned, a H.323 terminal is calling a SIP UA.
2. SHT sits between them again and accepts the call on SIP UA's behalf.
3. On receiving the SETUP request from H.323, the SHT forwards an INVITE signal to SIP. But unlike previously, SHT does not have the media capabilities of H.323 terminal. So, it includes a dummy SDP or no SDP.
4. On receiving, OK response from SIP UA, SHT sends a CONNECT signal to H.323 terminal.
5. As the INVITE response from the SIP UA contains its media description, SHT uses it to send and acknowledge H.323 capability negotiation and logical channel messages.
6. Once the acknowledgements for all the logical channel messages are received, the SHT knows the media transport address of H.323 endpoint and it can send re-INVITE with new SDP (actual session description of H.323 terminal) to SIP endpoint.

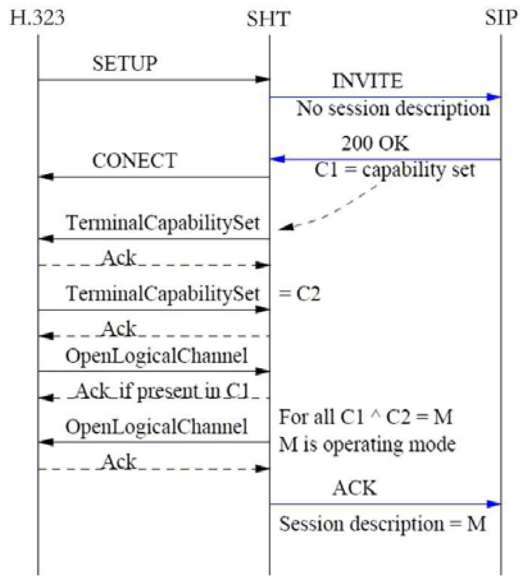

Figure 9: Call from H.323 terminal to SIP UA [12]

6. Analyses of Different Approaches

Now, we will try to analyze the aforementioned approaches. As most of the solutions presented until now are gateway based solutions, my comments are applicable to all gateway based approaches. Although, the solution presented in Sec. 5.1.1 seems flexible, it has some drawbacks such as:

- A H.323 gatekeeper has to store all the registrations of the SIP UA's it wants to reach.
- In this approach a H.323 gatekeeper can only know the SIP addresses handled by a typical registrar available to that H.323 zone. So, each H.323 zone would have to have an associated SHT.

Now, if we look at the solution presented in Sec. 5.1.2, it also has the similar drawbacks as the previous approach, namely that a SIP proxy has to store all the registration information of H.323 users, it wants to reach. But this approach does not require that every gatekeeper has to be equipped with a SHT. As long as at least one H.323 gatekeeper exists with a SHT, the SIP user can be located from the H.323 network. Although, the solution presented in Sec. 5.1.3 works well, it does not specifies how many SHTs are needed in a typical network, and also this method introduces some latency. Many of the vendors have already implemented gateway based solutions for SIP and H.323 interoperability such as: CISCO, SOUNDWIN etc. But, we think there are some problems with these gateway based approaches. Although, gateways provide us interoperability based on connectivity, the users often enjoy only those services that are common to both of the two networks. Which means, although the users were promised of a homogeneous network experience with the help of a gateway, they actually experience a very strict heterogeneous network. We think the solutions should be based on the convergence of these two protocols on higher levels of TCP/IP, say at application level. But, to provide application level convergence, the services and applications have to be separated logically from the device interfaces. In [15] the authors have presented another approach based on the client/ server architecture. Although they have focused the solution for IP video, this solution can be expanded to provide all the services. The device that provides this interoperability is named as VCON Media Xchange Manager (MXM). It integrates a suite of client/server applications. Users can experience real-time, interactive visual communications across the enterprise. They claim that MXM provides ease-of-use and advanced management capabilities. Although the architecture seems complicated, we think MXM is able to provide better interoperability between SIP and H.323 if expanded skillfully.

7. Conclusions

From this study we can conclude that, both of the protocols are somewhat similar in-terms of functionality and services they provide. H.323 has an elaborate and specific definition of supplementary services. All other scenarios are also well considered in H.323 definition. So, very few interoperability issues are expected to rise in case of H.323. Both, of the protocols are more or less the same in QoS support, as they have similar call setup delays and other issues that affect QoS. SIP, on the other hand, is flexible in case of adding newer features. Debugging and implementation is also easier. Both, SIP and H.323, are grasping features from one another in newer versions. So, we expect to see a completely interoperable SIP and H.323 versions in the near future. [17][1][6][11][9][4][5][8][14][2][12][10][13]

References

[1] Christian Stegh. How is SIP winning favor over H.323? http://searchvoip.techtarget. com.au/articles/18399.

[2] Fine - Tuning Voice over Packet Services . http://www.protocols.com/papers/voip2.htm.

[3] H. Schulzrinne, J. Rosenberg. A Comparison of SIP and H.323 for Internet Telephony. In *International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pages 83–86, July 1998.

[4] H. Schulzrinne, J. Rosenberg. The Session Initiation Protocol: Internet-Centric Signaling. In *IEEE Communications Magazine*, pages 134–141, October 2000.

[5] Henning Schulzrinne and Jonathan Rosenberg. Internet telephony: Architecture and protocols an IETF perspective. In *Computer Networks and ISDN Systems*, volume 31, pages 237–255, February 1999.
[6] IP Multimedia Communications. http://www.packetizer.com/ipmc/.
[7] Ismail Dalgic and Hanlin Fang. Comparison of H.323 and SIP for IP telephony signaling. In *Photonics East*, September 1999.
[8] James Toga and Joerg Ott. ITU-T standardization activities for interactive multimedia communications on packet-based networks: H.323 and related recommendations. In *Computer Networks and ISDN Systems*, volume 31, pages 205–223, February 1999.
[9] E. S. M. Handley, H. Schulzrinne and J. Rosenberg. SIP:Session Initiation Protocol. RFC 2543, The Internet Engineering Task Force, March 1999. http://ietf.org/rfc/rfc2543.txt.
[10] MBone Conferencing Applications. http://www-mice.cs.ucl.ac.uk/multimedia/ software/.
[11] K. Singh and H. Schulzrinne. Peer-to-peer internet telephony using SIP. CUCS 044-04, Department of CS, Columbia University, NewYork, NY, Oct 2004.
[12] K. N. Singh. *Reliable, Scalable and Interoperable Internet Telephony*. PhD thesis, COLUMBIA UNIVERSITY,2006.
[13] SIP Protocol ToolKit. http://www.radvision.com/Products/Developer/Protocol/SIP/.
[14] S. Taylor and L. Hettick. H.323 vs. SIP. *NetworkWorld Convergence Newsletter*, March 2002
[15] MXM: SIP Support. Delivering SIPU˝ H.323 Convergence and CoExistence Using a Video PBX Architecture. http://www.packetizer.com/ipmc/.
[16] N. Networks. A Comparison of H.323v4 and SIP. S2 000505, 3GPP S2, Tokyo, Japan, Jan 2000.
[17] Ismail Dalgic and Hanlin Fang. Comparison of H.323 and SIP for IP telephony signaling. In *Photonics East*, September 1999.

Authors Profile:

Shifta Ansari received B.Sc. in Computer Science and Engineering degree from Ahsanullah University of Science and Technology (AUST) in 2004. Currently she is the Lecturer of the Department of Computer Science and Engineering at the Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh.

Kashif Nizam Khan is an M.Sc student in Security and Mobile Computing studying under theNordSecMob- Erasmus Mundus Program in Helsinki University of Technology, TKK. He received his B.Sc Engineering degree from the Computer Science and Engineering (CSE) Discipline, Khulna University in 2007. He has served as a Lecturer in the Department of CSE in Ahsanullah University of Science and Technology and Khulna University. His research interests include next-generation wireless networks and security.

Jinat Rehana is an M.Sc student in Security and Mobile Computing studying under the NordSecMob- Erasmus Mundus Program in Helsinki University of Technology, TKK. She received her B.Sc Engineering degree from the Computer Science and Engineering Discipline, Khulna University in 2007. She has worked as Software Engineer in the Dohatec New Media. Her research interests include wireless sensor networks, protocols and security.

Nusrat Jahan Lisa received B.Sc. in Computer Science and Engineering degree from Ahsanullah University of Science and Technology (AUST) in 2007. Currently she is the Lecturer of the Department of Computer Science and Engineering at the Ahsanullah University of Science and Technology (AUST), Dhaka, Bangladesh.

Shahriar Kaisar received B.Sc. in Computer Science And Information Technology degree from Islamic University of Technology (IUT) in 2007. Currently he is the Lecturer of the Department of Computer Science and Information Technology at the Islamic University of Technology (IUT), Gazipur, Bangladesh.