

SIP-H323: A Solution for Interworking Saving Existing Architecture

G. De Marco¹, S. Loreto², G. Sorrentino³, and L. Veltri³

¹University of Salerno - DIIE- Via Ponte Don Melillo - 56126 Fisciano(Sa) – Italy
Ph.: + 39 0974 824700, Fax: +39 0974 824700, gdemarco@unisa.it

²Ericsson Lab Italy – Via Madonna di Fatima, 2 - 84016 Pagani – Italy
Ph.: + 39 081 5147733, Fax: +39 081 5147660,

salvatore.loreto@eri.ericsson.se

³Coritel - Via Anagnina, 203 - 00040 Roma – Italy

Ph.: + 39 06 72589169, Fax: +39 06 72583002,

{[@coritel.it">sorrentino,veltri](mailto:sorrentino,veltri)}@coritel.it

Abstract. In the 3rd generation multimedia communication world and in the 3GPP standardization consortium, SIP protocol appears to be the preferred signaling protocol. However, the need to communicate with non-SIP based network, e.g. H.323 from ITU-U, is still a reality. The need can be satisfied with the introduction of network gateways (also named Inter-Working Function). One of the open issues about SIP-H.323 interworking is the address resolution, in other words, the automatic forwarding of a SIP call to an H.323 user. The paper proposes a SIP network architecture which can interoperate with H.323 networks, safeguarding the existing software/hardware components, as SIP terminal clients or SIP server proxies or IWF gateways.

Keywords: Sip, H323, interworking, gateway, call routing

1 Introduction

One of the problems arising in the future multimedia network is the interworking between networks that use different protocols; at present days, such problems mainly concern the interworking between SIP (developed in IETF) and H.323 (from ITU) based multimedia networks. Both protocol are signaling protocol, being currently H323 the standard for any IP based implementation of multimedia communications ([1],[2],[3]). The 3GPP has selected SIP as the signaling protocol for multimedia communications in the UMTS network. All these considerations lead to the conclusion that the interoperation of H.323 and SIP based networks is becoming a very crucial problem ([5], [6], [7]). Among the various problems that arise when considering the interworking of these two protocols, one important aspect is to allow, for example, a SIP user to reach a remote user on both SIP and H.323 networks; of course if the remote terminal is an H.323 terminal, then an interworking system (gateway) is needed. A satisfactory solution, involving additional protocols [4]. In this work, we propose a new interworking solution that requires no modifications of these network elements. The proposed solution is so based on the assumption that neither the client applications (the terminals) nor the network servers/gateways should be

modified. Let us consider for example the following scenario: the owner of a big SIP network (> 500 consumers) has already acquired all the necessary servers; he/she has already installed and configured all the multimedia terminals; moreover, he/she has acquired the network nodes/servers and the H.323/SIP gateways (in the following referred also as *interface module* or *Inter Working Function*). Modifying the gateway source code or asking for a new version should be too expensive. In this context, we will see how to solve the addressing and registration aspects of the interworking problem without implementing the TRIP protocol within the SIP and H.323 signaling servers.

The main idea consists of the introduction of a new network component that easily allows the call forwarding from SIP to SIP domains or from SIP to H.323 domains.

2 System Outline

In a pure SIP network, terminals are named UserAgents (UA), while the servers can be classified as SIP Proxy servers, Redirect servers, and Registrar servers [8]. In our scenario, we consider a SIP network composed of UAs, stateful SIP Proxies acting also as Registrar servers, and one or more gateways (GWs) to other non-SIP IP networks. In such a network scenario, SIP terminals communicate directly with other SIP terminals and via an appropriate GW with non-SIP terminals. SIP Proxy servers are used to route call signaling among SIP terminals, by querying an internal database (DB). If the DB query gives no match for the current callee address, or if an error on the resulting next hop (SIP proxy) is obtained, the proxy releases the call and sends a *Not Found* message back to the caller. This fact may happen also and particularly in presence of a non-SIP called terminal; what really happens is that although the callee receives the SIP setup message, it is unable to generate an appropriate SIP response.

In order to forward call setup requests from a SIP based terminal (UA) to a H.323 user, a gateway entity (IWF) should include all the interworking functionality needed to translate transparently the SIP messages to H.323 signaling and vice versa. To make the correct forwarding of calls through the IWF possible, the client agents of the signaling servers (i.e. the SIP servers and the H.323 gatekeepers) should share some information about the presence of users behind the specific IWFs. Such information should be dynamically exchanged among the SIP servers, the IWF, and the gatekeepers. Although this action is not crucial between gatekeepers and gateways in an H.323 domain, there is not a straightforward solution for the SIP-to-IWF relationship, and a sort of specific protocol seems to be required.

A proposed solution for this issue makes use of the TRIP protocol. Another solution could be based on the adaptation of the SIP protocol and the change of SIP proxy functionality. However, both solutions seem to be not very suitable and won't be followed.

A possible approach that could be used to solve this issue is the insertion of a new module (software or hardware) acting as a SIP proxy server. This module should forward all the calls coming from SIP terminals to both the next hop SIP server and the SIP-to-H.323 gateway (IWF). However, the main drawback of this approach is that it requires the duplication of call signaling for both SIP-to-SIP or SIP-to-H.323 calls. This solution is fast but it increases (duplicates) the signaling traffic sent through the IP network.

An improved solution could be obtained by starting a new call request at the SIP proxy server as soon as a “Not found” message or a “Time out” message has been received. The new calling process is performed towards the preconfigured IWF. This solution decreases the signaling traffic, but increases the call setup time (up to $2T_{out}$, where T_{out} is the time-out for the SIP call). To be noticed that the proposed solution is a compromise between the increase of signaling traffic and call setup delay.

3 The Network Architecture

To describe the network architecture, let us start observing what happens if in a pure SIP network, a SIP user addresses a call for a user that is in an H.323 network. We suppose that the caller and called users are respectively a SIP user, whose address is *sip:amalfi@a_sip.com*, and an H.323 user, whose address is *positano@b_h323.com*. The address of the gateway is: *sip:gw.a_sip.com*.

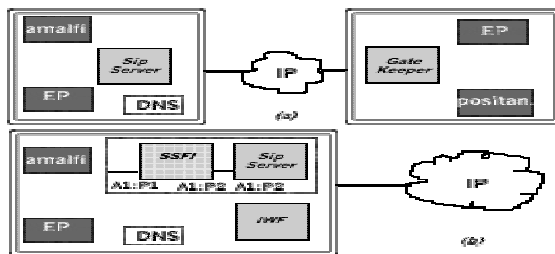


Fig. 1. (a) SIP-H323 standard interworking architecture(EP: end-point); (b) simplified structure of modified network

The user *amalfi* send a SIP INVITE message to the pre-configured SIP proxy server. As soon as the SIP Proxy Server receives the INVITE message, it tries to resolve the address contained in the field *To* of the message, consulting its “contact database” or by means of the DNS. If it cannot find any correspondence in the “contact database” for the user, it forwards an INVITE message to the *b_h323.com* domain; however, the H.323 domain cannot process successfully the message. Then the SIP Server answers the INVITE request by sending a *404Not Found* error message or a *408Request TimeOut* to the *amalfi* user. The previous result occurs even if an IWF is introduced in the SIP architecture (fig. 1). This is due to the fact that there is no mean to let the SIP server aware of the correct route of the SIP requests through the gateway. In other words, a call initiated by a SIP client and directed to a H.323 user would give negative results because of the fact that the SIP Server doesn’t know that it could address the call via the IWF. A possible solution proposed by the IETF is to register the IWF at the SIP Server using the TRIP protocol [4]. But even in this case it is necessary to have a new SIP Server in the network which is aware of the IWF and able to interpret the TRIP protocol.

By deeply examining the previously described scenario, it is possible to observe that the call failure towards an H.323 user is translated in a *404Not Found* or *408Request TimeOut* error message that is first received by the SIP server and then

forwarded to the caller (*sip:amalfi@a_sip.com* in the previous example). Noticeably, when receiving these error messages, the server might guess that the called user belongs to the H.323 domain and try to forward the call to the IWF.

This consideration is the basis of our scenario in which a SIP call that cannot be forwarded to the called user within the SIP domain is relayed through the IWF to the H.323 domain. For this scope, a new software component has to be introduced, the SSFI (Sip –Server –Functional to Interworking). The SSFI can be seen as a very simple and stateless SIP proxy server that just forwards all incoming messages (both requests and responses). The only functionality that it implements is to look for *404Not Found* or *408Request TimeOut* error messages and, when one of these messages is received, to translate them in a *302Moved Temporarily* redirection message with the address of the IWF in the *contact* field. Any other message that will arrive to the SSFI, will be just forwarded to the client.

In order to minimize the impact on the original architecture, the new SSFI can be introduced simply by configuring the SIP terminals to let them use the SSFI as the default proxy. As an example, if a terminal uses as its default outbound proxy a SIP server at the address $A1:P1$ (Address:Port), when using the new SSFI module, the latter is configured to accept messages on $A1:P1$, while the original SIP server will accept messages at $A2:P2$. If SSFI should run on the same machine as the SIP proxy server, then $A2 \equiv A1$ and $P2$ is one of the ports available on the server. Obviously the SSFI should forward every incoming call (from SIP user clients) towards the original SIP server using the socket $A1:P2$ (or $A2:P2$). We suppose that the SSFI and the SIP proxy are running on the same system.

The message flow between the SIP nodes is as follows: an INVITE message sent from the caller reaches the SSFI, the SSFI forwards it to the SIP server; if the SSFI doesn't receive a *200Ok* or *404NotFound* message within a time t , it starts trying to route the call towards the H.323 network by means of the IWF. We set this time t to $T_{out}/2$ (note that this isn't the optimal choice) [8].

4 Temporal Diagram

A client in the home SIP network sends an INVITE. The client asks *positano@b_h323.com* to establish a two-party conversation. The SSFI accepts the INVITE request and forwards the request to the SIP proxy server.

Both the SSFI and the Proxy Server set a Time-out counter. When the SIP proxy counter reaches the maximum value (T_{out}), the INVITE request is canceled. If the SIP proxy finds the called user before $T_{out}/2$, the terminal will send a *200 Ok message*.

If a *404 Notfound* message is sent to the SIP proxy before $T_{out}/2$ then the SSFI begins a new calling process in an other network using the gateway. The SSFI does not forward this response, but replies to the caller with the status codes *301 (Moved Permanently)* or *302 (Moved Temporarily)* specifying the IWF location with the Contact field. The caller then sends a new INVITE request to the SSFI with Request-URI set to the address specified in the Contact field.

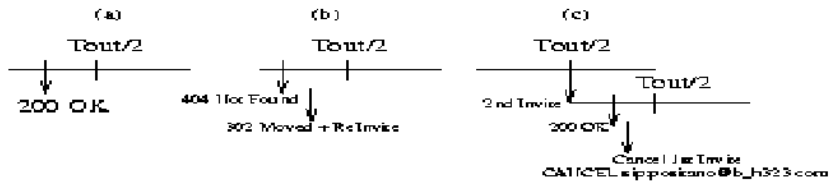


Fig. 2. (a) Successful transaction at SSFI; (b) Not Found in Sip network;(c) a successful response from the H.323 side, after the expiry of the first $T_{out}/2$

If no messages arrive to SSFI in a $T_{out}/2$ time, it starts a parallel search in the H.323 network. The SSFI then sends a new INVITE request to the SIP proxy with the same *To* (including tags), *From* (including tags), *Call-ID*, *Cseq* fields, but with a different Request-URI. Then it resets the Time-out counter. The Request-URI of the INVITE request is set to the IWF URI. For the SIP Proxy this request corresponds to a new transaction, and it should be proxied.

The “branch” parameter, in the new INVITE, is set to a different value. Actually this token must be unique for each distinct request. The SSFI uses the value of the “branch” parameter to match responses to the corresponding requests. CANCEL and ACK requests must have the same branch value as the corresponding requests they cancel or acknowledge. In this state, if a “not found” message arrives from the SIP network within $T_{out}/2$ seconds, the SSFI will keep on staying in a “wait” state. If a “not found” message arrives also from the IWF, SSFI will forward it to the SIP proxy, which will close the session. If a *200 OK* message arrives from one of the two networks, the SSFI will forward it as usual and, if necessary, will send a CANCEL message to the other network. The CANCEL message must be sent if a positive response arrives during the next $T_{out}/2$ seconds.

Just as an example, if we suppose that a *200 OK* response arrives from the IWF within the next $T_{out}/2$ seconds, the SSFI must send a CANCEL message to the SIP proxy. If neither the *200 OK* message nor the “not found” message should arrive from one of the two ways, the SIP proxy server will close the session, after $3/2 T_{out}$.

We do note that, if a *200 OK* message arrives from the IWF, it is possible to update the DB of the SIP proxy server in order to route future calls addressed to the called user, directly to the IWF. The SSFI could make this updating, sending special REGISTER messages to the SIP proxy.

5 SSFI: State Machine

The idle state of SSFI is T (Transparent). When the SSFI receives an INVITE message, its state changes to S (SIP context), and its counter is set. In S state, when a *200 OK* arrives from the SIP network, the SSFI goes back into T state; otherwise, when a *404 Notfound* message arrives, the SSFI goes into H state (H.323 context).

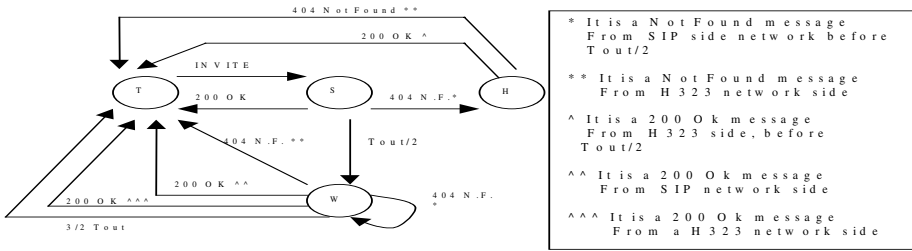


Fig. 3. State Machine

In the H state, SSFI begins a new session sending a “moved” message to the caller. When either a 200 OK or 404 Notfound message is received the SSFI goes back to the idle state T. Furthermore when in S state, after T_{out}/2, SSFI reaches the W state. In this state (Waiting) if a 404 Notfound message arrives from the SIP network, the SSFI continues waiting for some responses from the gateway.

6 Conclusions

In this paper the problem of the interworking between SIP and H.323 networks has been considered. The problem of call forwarding through different domains arises for calls generated from a SIP domain and directed to a H.323 domain. A possible simple solution has been proposed and described, taking into account particularly the problem of backward compatibility with previously installed SIP and H.323 networks and legacy systems. For this reason, the proposed solution does not use new protocols between signaling systems and does not require any modifications of SIP/H.323 terminals nor servers. The call can be forwarded to both domains in serial or parallel manner. A compromise is chosen in order to balance the generated signaling traffic and the call-setup delay.

References

- [1] “Packet based multimedia communication systems”, Recom. H.323 – ITU-T, Feb. 1998
- [2] “Call Signaling Protocols and Media Stream Packetization for Packet Based Multimedia Communications System”, Reccom. H.225.0, Version 2 - ITU-T, Feb. 1998
- [3] “Control protocol for multimedia communication”, Recom. H.245.0, ITU-T, Feb. 1998
- [4] J. Rosemberg, , H Salma, “Usage of TRIP in Gateways for Exporting Phone Routes” March, 2000
- [5] Singh, Schulzrinne, “Interworking Between SIP/SDP and H.323” May 12, 2000
- [6] H. Agrawal, R. R. Roy, et Al. “SIP-H.323 Interworking Requirements”, February 22, 2001
- [7] H. Agrawal, R. R. Roy, et Al. “SIP-H.323 Interworking”, July 13, 2001
- [8] H. Schulzrinne, J. Rosemberg, et Al. “SIP: Session Initiation Protocol”, July 20, 2001