An Embedded Software Approach for the Development of SIP-Based VoIP Server

Roy Chaoming Hsu
Department of Computer Science and Information Engineering
National Huwei University of Science and Technology
rchsu@nhust.edu.tw

Cheng-Ting Liu, Wen-Ping Huang, Jun-Jay Yang
Department of Aeronautical Engineering
National Huwei University of Science and Technology
#64 Wen Hua Road, Hu-Wei, Yun Lin County 632, TAIWAN

Abstract

In this paper, embedded software engineering approach is employed for the development of a SIP-based VoIP server. Signaling control functions of SIP servers are integrated and implemented in an embedded system platform. This embedded SIP server also serves as network access server (NAS) client to a centralized RADIUS server to provide user’s authorization, authentication, and accounting information. Other advanced telephony features, such as voice conferences, call waiting, call on-hold, etc., can be designed in the embedded SIP server as well. In this study, the embedded software architecture and development procedure for the embedded SIP server such as hardware platform, the embedded software function, development procedures, and functional test are described. This embedded SIP server can be beneficial for frequent computer users with larger telephone usage, such as people working in global branch offices of enterprises, schools on the academic communication backbone, or research institutes among Universities on an academic broadband infrastructure.

Key words: Embedded Software System, Internet Telephony, Session Initiation Protocol, Real-Time Transport Protocol

1. Introduction

SIP (Session Initiation Protocol, RFC 3261) [1] is a text-based and HTTP-like application-layer signaling control protocol for establishing, modifying, and terminating multimedia sessions, such as Internet telephony calls, multimedia conferences, and similar applications with one or more participants. SIP is a new generation of Voice over IP (VoIP) signaling protocol, which relies on RTP (Real-Time Transport Protocol) [2] and RTCP (RTP Control Protocol) for multimedia data exchange and control. Due to the vast volume of signaling in Internet Telephony system, standard SIP-based VoIP system categorizes the signaling functions into different servers and assigns each function to different server PC. Owing to the advance of CPU and SoC (System-on-Chip) technology, using embedded system with high performance CPU to design and develop information and communication products are becoming mature and pervasive. Thus, integration of the various SIP-based server functions into a single embedded SIP-based server is becoming possible. Other advanced features of Internet Telephony such as pre-paid billing, voice conference, unified message services, etc., can also be implemented into a single embedded system. In this paper, a SIP-based VoIP server is implemented using embedded system and the implementation results will be described. Section 2 will describe the related work for the Internet Telephony. Software functions of the development of embedded SIP server will be described in Section 3. In Section 4, the system implementation and function testing are presented. Section 5 is the
development and testing results. Section 6 comes out the conclusions and further work.

2. Related Work

Initiations of SIP protocol design and SIP VoIP research was proposed by H. Schulzrinne and J. Rosenberg [3] in 1998. In the next year (1999), the RFC for SIP was formally proposed by IETF and published in IEEE Network [4]. Since then numerous research work has been focus on the SIP based Internet Telephony Architecture and system. Bellcore has used Java Telephony API and SIP to design telecommunication system called ChaiTime [5], which was based on PBX (Public Branch exchange) system and can be used to integrate with other telecommunication service. Gbaguidi of the Swiss Federal Institute of Technology established a VoIP telecommunication system [6], which integrated different Java Bean package for the VoIP applications. The VoIP system developed by Cornell University, called ITX [7], is similar to the Gbaguidi’s, which also use the mix-and-match strategy to design the VoIP telecommunication service. Different with Gbaguidi’s is that the ITX adopt the “Template” model to develop a series of high level Java API such that the developers of the VoIP system can re-use the software component to integrate a VoIP system which fit the individual’s requirement. Integration of VoIP service with wireless network is only recent. Rao, Lin, and. Cho [8] have explored the feasibility of integrated VoIP service into the existing GSM system, which is called iGSM. In their study, the authors have investigated how to use H.323 terminals or the GSM handsets to access VoIP service and to realize the registration, voice data transmission, and teardown without changing the current GSM system. Recently, SIP-based VoIP systems with new and innovative services are proposed [9, 10, 11]. A SIP-based VoIP system with basic function of multi-point call, multiple audio coding formats and complex functionalities such as call muting, call-hold and personal mobility support is proposed and developed by S. Zeadally, and F. Siddiqui [9]. R. V. Prasad, R. Hurni, and H. S. Jamadagni [10] considered the limitations of existing SIP conferencing methods and proposed a distributed architecture using Controller (SIP Proxy Servers) and Conference Servers which facilitates the control and media handling of a VoIP conference. Another novel software architecture providing the connected multimedia services through networked embedded systems using SIP protocol are proposed by J. Y. Kwak, et al. [11].

By reviewing the related research, one can see that, concerning the vast volume of signaling transactions in a realistic telephony system, most of the above-mentioned SIP-based VoIP system categorizes the SIP signaling control function into different servers and assigns each function to different server PC. Even though some literature integrated partial function of SIP servers into a general PC, while none of them integrated all the signaling functions into a single SIP server. Since VoIP is the product of information technology and Internet era with the advantage of cost saving and convenient to the computer users, early deployment of Internet Telephony should first consider the frequent computer users with larger telephone usage, such as people who are working in global branch offices of enterprises, schools on the academic communication backbone, or research institutes among Universities on an academic broadband infrastructure. Providing an integrated SIP server using embedded system for specific end-to-end groups of frequent computer users should be the primary concern for pervasive Internet Telephony deployment. Implementation of SIP server functionalities on an embedded system provides the basic operating unit of the last mile for the Internet Telephony infrastructure. By incorporating the functionalities of embedded SIP server software into the routers, added-value of Internet Telephony can be offered by the Internet service and equipment providers as well.

In this study, a SIP server which integrated the various function of SIP servers utilizing embedded system and targeting the above-mentioned communities are proposed. The detail design and implementation is described next.

3. Software Functions of the Embedded SIP Server

In this section, functions of SIP-based signaling for handling the setup, modification, and teardown of multimedia session will be introduced. These functions also constitute the embedded software requirement for the development of the SIP-based VoIP server.

SIP Overview

SIP is a signaling protocol that handles the setup, modification, and teardown of multimedia sessions. It is believed that SIP is becoming the dominant VoIP signaling protocol because SIP is simple, more flexible and suitable for supporting intelligent user devices and for the implementation of advanced features. Computer
hosts or communication devices can be implemented with the appropriate SIP functionalities or services according to the role they play in the Internet Telephony system. The SIP-based VoIP system can be deployed as three basic network topology, which are end-to-end, local LAN, and Internet or WAN. SIP defines two basic classes of network entities: clients and servers. A client, (also known as a User Agent Client, UAC) is an application program that sends signaling requests. Four different types of servers can be defined according to the network topology, which are Proxy Server, Registration Server, Location Server, and Redirect Server.

The function of SIP signaling basically can be categorized into request and response, which are implemented in the application layer of the existing Internet protocol stack. RFC 3261 defines 6 types of request methods: INVITE, REGISTER, CANCEL, BYE, ACK, and OPTIONS. Response can be categorized into one provisional and 5 finals, while finals are also classified into two corrects (success and redirection) and three failures (request, server, global failure). The address in SIP is similar to the Email address (username@hostname), which is called SIP URI (Uniform Resource Indicators). SIP message consists of start-line, message-headers, and message-body. The SIP message expressed in BNF form is shown below:

```
SIP message = start-line *message-header CRLF

start-line = request-line | status-line

message-header = [message-body]

start-line = request-line | status-line

where request-line = method SP request-URI SP SIP-Version CRLF,

status-line = SIP version SP status code SP reason-phrase CRLF,
```

The meaning of the symbols are explained as below:

- *: represents at least one line
- CRLF: represents control and line feed
- SP: space

**SIP Servers**

SIP use RTP (Real-Time Transport Protocol) to transmit the voice data, which is appropriate for peer-to-peer transmission of real-time multimedia. Implementation of RTP is constructed atop UDP and employs RTCP (RTP Control Protocol) to monitor session quality and to detect network problems. After SIP Server finishes signaling between caller and callee, the session data will transmit between caller and callee using RTP without going through the SIP server. The basic functions of SIP-based VoIP server are Proxy Server, Location Server, Registration Server, and Redirect Server, which functions provide the embedded software requirement for the embedded SIP server, as listed below.

1. **Proxy Server**: Proxy server provides logical routing for the URI requests from User Agent Client such that the message can be transmit to the expected destination. Certain commercial applications such as pre-paid billing, network quality control, etc…. can be realized using a Proxy server.

2. **Location Server**: Location server uses dynamic database to store address of UAC.

3. **Registration Server**: Registration server accepts the address registration, updates request from UAC and stores the updated data to the Location Service.

4. **Redirection Server**: The main function of Redirection server is to search into the Location Server for the updated address and reply to the request for further processing.

The embedded SIP server system developed is shown in Figure 1, where the embedded SIP Server A and B are two of the embedded SIP servers which supports up to 253 users in a LAN.

![Figure 1 The developed embedded SIP server system](image)

Because the SIP server function is build in the application layer, the protocol reserves flexibilities for further advanced feature planning to suit the networked embedded software environment such as unified message services, voice conference, call waiting, billing, etc…. As shown in the Figure 1, integrating with the functions of Registrar, Location server and Redirect server, the embedded SIP server also act as network access server (NAS) and will provide users’ authorization, authentication, and accounting information to a centralized RADIUS...
server in the Internet. In the next section, the implementation of embedded SIP-based VoIP server which integrate the signaling functions of SIP server into the selected embedded system development platform will be discussed. The installation of a UAC software in users' PC or notebook for testing voice communication between UACs will also be demonstrated.

4. System Implementation

In this section, the embedded system platform, the development tools and environment, and the software architecture for the implemented embedded SIP server will be discussed.

Embedded Platform and Development Tools

To design and implement the embedded SIP server using embedded software engineering approach, an embedded system platform should be selected first and the development environment be established. The selected CPU of the embedded system platform is a 32-bit ARM7 network processor, which is a high efficiency and low cost 32-bit RISC processor and is designed for the network application with low power consumption. The bootloader for the embedded system platform is provided by the vendor whose main function is to initialize memory, register, I/O, do memory re-mapping, invoke the operation system (OS), and finally turn over the control to the OS. The OS used in the embedded SIP server is uCLinux, which is based on Linux 2.4 kernel and is suitable for the non-MMU processor. The development environment used for this implementation is CYGWIN, Domingo IDE and WINeZ ARM ICE (In-Circuit Emulator). CYGWIN is a UNIX emulator under Microsoft Windows PC, which provides a convenient development environment for PC users. Domingo is an integrated design environment (IDE), provided by Microtime Computer Inc, which can perform single step debugging function when incorporated with the WINeZ ICE. The system block diagram for designing and debugging the embedded SIP VoIP server is depicted in the Figure 2 as follows. The serial port in the figure is to provide a console for the development PC, while the JTAG I/F are for the WINeZ ARM ICE.

Figure 2 System block diagram for designing and debugging the embedded SIP-based VoIP server

During the early design and development phase, a Pentium IV PC with Linux Mandrake 9.2 was used to design the SIP server related application programs and to perform the compatibility test before porting and integrating the software to the ARM-based embedded platform. The SIP related application programs were then cross-compiled in the CYGWIN of the Windows 2000 PC and transformed into an executable image file (*.elf) for burning into the ROM of the embedded platform.

The Software Architecture

Protocol stack for the implemented SIP-based VoIP server can be seen in the Figure 3. The functionalities of the SIP server are implemented in the application layer by using the underlying UDP in the Transport Layer.

Figure 3 Protocol stack for the implemented SIP-based VoIP server

The SIP Server will invoke system call sendto() and pass the transmitted parameters into the function. sendto() will use the Linux socket to pass data to underlying layers and transmit into the physical network. On the receiving side, the SIP Server will invoke recvform() system call to receive incoming UDP packet. In the implementation, both the SIP
Server and the UAC are designed using the sendto() and recvfrom() system call, the system calls used and their functionalities are listed in Table 1.

Table 1 Listing of the system calls and their functions used in implementing the SIP Server.

<table>
<thead>
<tr>
<th>system call</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>socket()</td>
<td>Open socket</td>
</tr>
<tr>
<td>close()</td>
<td>close socket</td>
</tr>
<tr>
<td>bind()</td>
<td>Set host IP and port</td>
</tr>
<tr>
<td>listen()</td>
<td>accept connect command</td>
</tr>
<tr>
<td>connect()</td>
<td>confirm connection</td>
</tr>
<tr>
<td>accept()</td>
<td>make a new socket as accepted</td>
</tr>
<tr>
<td>recv()</td>
<td>receiving message</td>
</tr>
<tr>
<td>recvfrom()</td>
<td>receiving message (for UDP)</td>
</tr>
<tr>
<td>send()</td>
<td>sending message</td>
</tr>
<tr>
<td>sendto()</td>
<td>sending message (for UDP)</td>
</tr>
</tbody>
</table>

Flowchart for the functionalities of REGISTER and INVITE implemented in the embedded SIP server software is shown in Figure 4.

![Simple SIPServer flow chart](image)

Figure 4 Flowchart for the implemented embedded SIP server software

The program will starts by initializing and opening UDP socket and enter main program. In the main program, the User information database will first be initialized and will be read into the user’s data structure for further query and the main program will wait for request from the network. When request comes, the software will parse the incoming request method. If the incoming request is not INVITE or REGISTER, the software will execute the error process to reply with error response (4XX error) and return to the waiting-for-request state. The software will execute either INVITE or REGISTER process once the incoming request is INVITE or REGISTER and generate the response for transmitting back into the network accordingly. The software will return to the waiting-for-request state after the response is send and wait for the next SIP request.

5. Development and Testing

In order to test the functionality of the implemented embedded SIP server, a User Agent Client (UAC), programmed in Java, was designed for the hosts of the caller and callee. The designed simple UAC for testing the embedded SIP server can be shown in Figure 5. Users of the UAC can initiate the request message by clicking the service icons, and the response message from the embedded SIP server will be received and displayed on the UAC. The simple UAC can be expanded to incorporate other multimedia functions, such as ringing tones for different users on the address book, display of the callee’s image, etc…

![Simple UAC for testing the embedded SIP-based VoIP Server](image)

Figure 5 Simple UAC for testing the embedded SIP-based VoIP Server

The symbols used in the UAC is explained as the following,

- 🔄: REGISTER icon
- 💬: INVITE icon
- 📞: RING and HANGUP icon

The embedded SIP-based VoIP Server software designed in this project is named SIPServer with the current version X1.5. To test the SIPServer, the following devices in a LAN environment are required:
An 8 ports switching HUB
2 PCs (or Notebooks) or more with Ethernet, microphone/speaker and with Java Runtime Environment (JRE) installed
CAT 5E RJ-45 cables

The system layout for the testing configuration can be shown in Figure 6.

Figure 6 Configuration for testing the SIPServer
Embedded system in a LAN environment

The designed UAC will be installed to the hosting PCs to test the functionalities of the Registration Server, Location Service and Redirect Server. The testing procedures are described as below

At Caller’s end:

Step 1 : Registration
Caller registers by filling the name and IP address in the UAC and click the Register icon. The response message of registration return from the SIPServer will be displayed in the center blank of the UAC.

Step 2 : Invitation
Once registered, the caller selects the callee from the address book of the UAC and click INVITE. The SIPServer will invoke the Location Service to find out the callee’s URI and initiate the ring process, while the INVITE message will be displayed in the caller’s UAC. When seeing the vibrating ringing icon, the callee and caller can click on the ringing icon to invoke the RTP session software GUI, which is shown in the Figure 7, to start the voice data communication.

Figure 7 RTP session software GUI

At callee’s end:

When an INVITE message has been received from the user’s UAC, the RING icon will vibrate and the callee can decide whether to pick up the phone call or not. If the callee clicks on the RING icon to pick up the phone call, the RING icon will change to HANGUP icon and, at the same time, the callee will invoke the RTP session software GUI. By filling in the IP address of the caller and click on the connect the voice communication can be initiated until the EXIT icon in the RTP GUI is clicked to HANGUP the call.

The embedded SIP Server has been tested in a computer networking Lab with 40 PCs for the scalability test. The 40 PCs were initiating the REGISTER and INVITE simultaneously to each other, while the SIPServer was found working perfectly. The next step for the experiment will be to transport the SIPServer to Internet and employ automatic dialing software to perform simultaneous signaling test and to monitor the performance with UDP/TCP packet analysis and monitor software. Further testing on the embedded SIP server is under planning, which should include the interoperability test to other SIP-based Internet Telephony system and the security test, once the full SIP server functionalities are integrated and completed.

6. Conclusions and Further Works

In this paper, embedded software engineering approach of a SIP VoIP server on embedded system is described. The embedded hardware platform, software architecture, development procedures, arrangements for testing the embedded SIP server is conducted and implementation results are shown. The SIP server is implemented in an embedded system platform which utilized a high performance, low cost with low power consumption network processor. Integrated function of SIP servers and a User Agent Client for software testing are designed and the procedures for software porting from Linux PC to the embedded platform are also described. The function of Registration server, Location Service, and Redirect Server are executed correctly under a simultaneous signaling test for a computer network Lab with 40 PCs. Testing for larger number of callers and for the functionality of proxy server is undergoing. The SIP server can also acts as a network access server (NAC) client for providing authorization, authentication, and accounting (AAA) information to a centralized RADIUS server for billing and management purpose. Other advanced features of telephony system such as voice conference, unified message service, call waiting and pre-paid billing can also be integrated in the embedded SIP server. The
developed embedded SIP server is suitable for groups of frequent computer users with larger telephone usage, such as people working in global branch offices of enterprises, schools on the public communication backbone, or research institutes among Universities on an academic broadband infrastructure.

Implementation of SIP server functionalities on an embedded system provides the basic operating unit of the last mile for the client-server type of Internet Telephony infrastructure. By incorporating the functionalities of embedded SIP server software into the routers, added-value of Internet Telephony can be offered by the Internet service and equipment providers as well. However, the centralized client/server network of SIP-based Internet Telephony is facing a new threat from the technology of decentralized peer-to-peer (P2P) VoIP network. Skype [12] is a peer-to-peer VoIP technology, launched by former KaZaA funder Niklas Zennstrom and Janus Friis, which can penetrate firewall and NAT (Network Address Translation) and claimed to provide analog telephony voice quality. Skype offers free PC to PC phone call but charges for PC to PSTN and mobile phone. The impact of peer-to-peer VoIP network, such as Skype, to the centralized client-server SIP-based network might depend on the user’s habit and attitude and the business model the VoIP carrier provides, yet the results are still unknown for lacking the market survey for the time being. Future work should focus on analyzing the pro and con of the centralized embedded SIP VoIP server and the decentralized P2P VoIP network such that a better, cheaper and full-featured Internet Telephony system suitable for ordinary users can be deployed.

References


