Distributed VoIP System for Ad-hoc Network with Pseudo SIP Server

LIN-HUANG CHANG, PING-DA CHUANG, YU-JEN CHEN
Grad. Inst. of Networking and Communication Eng., Chaoyang University of Technology
168, Jifong E Rd., Wufong Township, Taichung County 41349, Taiwan

Abstract: - Traditional Voice over Internet Protocol (VoIP) technique is based on client/server architecture. In this paper, we proposed a VoIP architecture for Ad-hoc network that provides users to communicate with each other directly through embedded pseudo SIP server. This architecture will establish a distributed VoIP system. The performance analysis of signal transmission delay will show the advantage of the proposed distributed VoIP system compared with the standalone infrastructure.

Key-Words: - VoIP, pseudo SIP server, distribution, blocking probability

1 Introduction
Voice over Internet Protocol (VoIP) is one of the killer applications over Internet recently. VoIP integrates voice and other multimedia by using packet switch and reduces the cost or provides free charge when people request a long distance voice communication service. Traditional VoIP approach, such as Session Initiation Protocol (SIP)[12], employs the client/server architecture which however needs standalone SIP servers to serve clients.

The standalone SIP servers, including SIP registrars, SIP proxy and SIP redirect servers, will store current user location information, proxy call invitation procedures and reply caller information, respectively. When the call activities increase, the standalone SIP servers may suffer from overload and traffic bottleneck problems. Also, the maintenance of the centralized standalone SIP server will become one of the routine services.

The other approach opposite to the centralized VoIP architecture is the de-centralized or distributed design that solves the problem of traffic bottleneck. To conquer the maintenance issue, a light middleware SIP server embedded in each individual node will become a suitable solution. Such embedded SIP server is expected to be compatible with the existed user agent (UA).

Therefore, in this paper, we propose a distributed VoIP architecture for Ad-hoc nodes with embedded pseudo SIP server. The designed system allows user to advertise itself to all existing users in Ad-hoc network during bootstrap stage, and discovers the existing users’ information right after the advertisement. The proposed embedded pseudo SIP server handles remote user discovery procedures and acts as a middleware to intermediate between SIP UA and transport layer. Users can communicate with each other via the proposed embedded pseudo SIP server without particular standalone SIP server. This design will solve the bottleneck problem of standalone SIP server while serving massive client requests. In addition, the proposed pseudo SIP server architecture will reduce the signaling delay time compared with the centralized VoIP system. This paper will describe the novelty of the proposed pseudo SIP server and the advantages of the distributed VoIP system compared with centralized standalone architecture.

The rest of this paper is organized as follows. In section 2, we will discuss the researches related to the distributed VoIP system and the discovery mechanism of UAs. Section 3 will describe the proposed system architecture in detail. The performance analysis in terms of signaling handling delay will be carried out in section 4. Finally, we will conduct conclusions for this research.

2 Related Works
The research in [9] has proposed a framework for the Ad-hoc VoIP network. They modified the SIP protocol to provide remote user discovery and the capability of audio/video conferencing in a distributed VoIP system. The modified SIP protocol relied on broadcasting the discovery message to probe the existing users within the same domain. Each node identified itself by introducing a unique id to other nodes and kept track of others by storing the remote registration message.

The advantage of this research is to employ a simple approach for Ad-hoc VoIP service without dedicated standalone SIP servers. However, the modification of SIP protocol only works for certain SIP UA but may not be compatible to most SIP UAs. In addition, the possible broadcast storm for discovery under dense user environment will become another problem.
Another approach [10] on Ad-Hoc VoIP system was also discussed. This system, however, proposed two modes for implementation. The first mode employed a modification of SIP protocol similar to the research [9]. It therefore led to similar problems. The second mode introduced existing Service Location Protocol (SLP)[8], such as UPnP[14], Jini [7] and Salutation[2] for message exchange and remote user discovery to avoid possible broadcast storm in dense user environment. However the approach using SLP still suffered from the compatible problem because it relied on the integration of SLP with particular SIP UA. The tight coupling between SLP module and SIP UA may cause communication problem for SIP signaling. Therefore, the approach can not support the Ad-hoc VoIP completely.

Regarding the integration of discovery mechanism with UA, IETF has proposed an SIP extension [11,13] using REGISTER message to integrate with discovery mechanism in 2004. This additional message was described in XML format and attached in the tail of original REGISTER message. Although the extension provides an application to send extra message within REGISTER message via SIP standard protocol, it still relies on the support from UA to accomplish extra functionality. This will slow down the evolution of add-on features for VoIP applications.

It is therefore important to take advantage of distributed VoIP architecture with UA compatible by introducing a SIP functionality middleware integrated with the discovery mechanism in each node. In this paper, we propose the embedded pseudo SIP server as the middleware. This proposed VoIP architecture distributes the traffic to each node’s embedded pseudo SIP server rather than particular standalone server. This will solve the bottleneck problem of standalone SIP server and reduce the signaling delay time. On the other hand, the proposed embedded pseudo SIP server will handle the remote user discovery procedures and act as a middleware to intermediate between SIP UA and discovery procedures. This will accelerate the promotion of advanced SIP functionalities or VoIP features. The detailed system design of the proposed architecture will be presented in the next section.

3 System Architecture and Design

The proposed VoIP system provides a configuration-free VoIP architecture under Ad-hoc wireless network. The test bed of our design allows each user to communicate with others without any information setting. There is no standalone infrastructure for network access in this design. The system architecture can be categorized in two types of view: the macro view and micro view.

The macro view of the system architecture is a complete set of network topology for VoIP services, such as network routing facilities, SIP servers, and SIP UAs. The macro views for standalone VoIP and Ad-Hoc VoIP architectures are shown in Fig. 1 and Fig. 2, respectively.

From the macro view of standalone VoIP architecture shown in Fig. 1, all SIP UAs acquire the VoIP services via regional SIP servers or SIP proxies. According to the original design of SIP protocol, a call invitation to remote network will be routed from designated default SIP server through infrastructural internetworking until it reaches the target network.

For the macro view of proposed Ad-hoc VoIP system, as shown in Fig. 2, the Ad-hoc internetworking will replace the infrastructural standalone internetworking. All UAs are connected via Ah-hoc mechanism within wireless network coverage. In our design, each UA contains an embedded pseudo SIP server and does not need any particular server for SIP signal redirection or proxy. Our design will accomplish a full Ad-hoc and distributed VoIP architecture. When a call is initiated to a remote network, the requested signaling
messages will be routed from proposed pseudo SIP server through Ad-hoc internetworking. In general, the SIP signaling processing is irrelevant to the network routing protocol. Therefore, the SIP signaling messages for standalone and Ad-hoc VoIP architectures can be simplified as micro view.

3.1 Pseudo SIP server
Former approach on Ad-Hoc VoIP system [9] modified the SIP protocol to provide defined functions. Such approach only works in certain SIP UA due to non-standardized messages. Our proposed pseudo SIP server takes the advantage of the modified SIP protocol design and in the mean while solves the UA compatible issue.

The protocol stack of proposed scheme is shown in Fig. 4. The embedded pseudo SIP server, which is an additional daemon running inside every mobile device, provides basic SIP functions including SIP registrar and SIP redirect/proxy servers for local requests. It is a light plug-in middleware which is responsible to handle SIP requests from upper layer SIP UA as well as requests from the remote users. The separation of pseudo SIP server from SIP UA provides the flexibility of the system design. The pseudo SIP server is kept compatible with SIP UA easily and provides various functionalities such as remote user discovery.

The addressing problem is a major issue for users under Ad-hoc network circumstance. Due to the lack of a centralized standalone server to offer an address prior to communication, such as DHCP service, we employ IPv6 in our protocol stack for addressing. The deployment of IPv6 allows user to obtain a unique link-local address in bootstrapping stage without a particular server. It provides auto-configuration and state-less mechanism on addressing.

The process flow of proposed distributed pseudo SIP server is shown in Fig. 5. During discovery phase, the pseudo SIP server will first collect other users’ information in the network domain. This can be done by using SIP REGISTER message. The detailed discovery mechanism integrated with registration will be discussed in the next sub-section.

When the pseudo SIP server receives the REGISTER message from its UA (user 1, as shown in Fig. 5), it broadcasts this REGISTER message. Other users (user 2 to user N) obtain the information of user 1 after receiving the REGISTER message and then reply 200 OK to pseudo server of user 1. Upon receiving 200 OK replied from different users, pseudo SIP server will establish the contact or buddy list.

During call setup phase, pseudo SIP server will play the similar role as standalone SIP proxy/redirect server. However, the transmission time between SIP UA and pseudo SIP server, which is embedded inside the mobile user, is expected to be reduced compared with the standalone model.

Fig. 4 Protocol stack of distributed VoIP system

Fig. 5 Process flow of distributed pseudo SIP server

Fig. 6 Process flow of standalone SIP proxy server
For comparison, we discuss the process flow of standalone SIP proxy server, shown in Fig. 6. Usually, the SIP proxy server may provide the DHCP service. Therefore, during discovery phase, user 1 acquires the IP address from SIP proxy server via DHCP four-way handshakes which is followed by sending a REGISTER message to SIP server for registration. User 1 can initiate a call after the completion of registration. During call setup phase, the standalone SIP server will be responsible for forwarding SIP messages between user 1 and other users.

3.2 Integrated Discovery Mechanism

We proposed an integrated discovery mechanism which allows user to advertise itself and discover other users by attaching additional information with standard SIP REGISTER message as shown in Fig. 7. The additional information is written by XML format which contains self user information such as SIP URI, current presence status as well as last available time, and so on. Once a user completes the addressing phase via state-less auto-configuration of IPv6 during bootstrap stage, it will multicast the REGISTER message to the user by using the defined IPv6 multicast address (FF0C::80) with port number 7000. The remote user who supports this mechanism will respond another SIP REGISTER message via unicast transmission. The discovery phase is completed upon receiving this replied REGISTER message which includes remote user’s information.

Because the SIP REGISTER message contains non-standardized specific information, thus we handle this message flow by proposed pseudo SIP server. The pseudo SIP server is capable of XML parser and reads the information in REGISTER message. The REGISTER message will be separated into two parts, including SIP standard and user description. The SIP standard part is responsible for SIP registration as usual. The user description, on the other hand, will deal with the discovery process. With the embedded pseudo SIP server, the proposed distributed VoIP system integrates the registration procedure with discovery mechanism.

Our proposed pseudo SIP server also provides the event subscription functionality under Ad-hoc VoIP scenario. After completing the discovery phase, the pseudo SIP server will initiate the event subscription to query remote users’ information, such as personal data, presence status and so on. If the requested information has been changed, the remote users will respond with up-to-date information. By employing such event subscription instead of periodical broadcasting, the proposed pseudo SIP server will reduce the unnecessary message flows and bandwidth usage, especially during the dense user situation.

The proposed event subscription extension follows the presence and event notify, defined in RFC 3856. It includes two commands: SUBSCRIBE and NOTIFY commands. The command SUBSCRIBE will initiate an event subscription for specified event alerted to subscribers by NOTIFY command.
The message format of SUBSCRIBE command is shown in Fig. 8. There is no extended description part in subscription command which is different from the discovery message. The event for this subscription will be described by the field “Event”, for example, presence status change or registration status change.

Once the event notification is triggered, the pseudo SIP server will initiate the notify phase by unicasting NOTIFY command with changed event information, shown in Fig. 9. The NOTIFY message contains two parts being the same as the discovery messages. The extended description part is described in “application/reginfo+xml” format. It also contains the requested messages related to this notification. The subscriber will conduct related action upon receiving the notification information.

4 Performance Analysis

For the analysis of signaling handling delay, we employ the SIP signaling message flows shown in figure 5 for distributed pseudo SIP server case and process flows shown in figure 6 for traditional standalone SIP server case.

For distributed VoIP mode, we perform a VoIP call from user 1 (UA1) to user 2 (UA2) within wireless Ad-Hoc network. All devices are connected via Ad-hoc mode wireless network running FreeBSD with IPv6 auto-configuration link-local addresses. In addition to the registration functionality, the SIP servers for both cases act as SIP proxy servers. Since the simplified model for micro view of VoIP systems will treat the message routing (hopping) delay as a constant value for both distributed and standalone architectures. Therefore, we will ignore this constant part for our subsequent numerical analysis.

In the following experiments, we will measure and compare the overall process delay of initiation prior to voice data transmission. The overall process delay consists of discovery delay and call setup delay. Theoretically, both discovery delay and call setup delay are composed of the command handling time and round-trip transmission time.

From Fig. 5, the total transmission time for the distributed pseudo SIP server, T\text{Dis}_\text{SIP}, can be expressed as equation (1).

\[
T_{\text{Dis}_\text{SIP}} = 5T_{\text{UA1}_\text{SIP}} + 5T_{\text{SIP}_\text{UA2}} \quad (1)
\]

The definition of the related parameters is summarized in Table 1. We assume the transmission time from UA 1 to SIP server, T_{\text{UA1}_\text{SIP}}, equals to that from SIP server back to UA1, T_{\text{SIP}_\text{UA1}}, and the same as UA2 case.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Standalone SIP Server</th>
<th>Pseudo SIP Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{\text{UA1}_\text{SIP}}</td>
<td>Transmission time from UA1 to SIP server</td>
<td>10ms</td>
<td>1ms</td>
</tr>
<tr>
<td>T_{\text{SIP}_\text{UA1}}</td>
<td>Transmission time from SIP server back to UA1</td>
<td>10ms</td>
<td>10ms</td>
</tr>
<tr>
<td>T_{\text{UA2}_\text{SIP}}</td>
<td>Transmission time from UA2 to SIP server</td>
<td>0.5ms</td>
<td>1ms</td>
</tr>
<tr>
<td>T_{\text{SIP}_\text{UA2}}</td>
<td>Transmission time from SIP server back to UA2</td>
<td>1ms</td>
<td>1ms</td>
</tr>
</tbody>
</table>

The overall process delay, T_{\text{Dis}_\text{Overall}}, including SIP and UA command handling time, for the distributed pseudo SIP server therefore can be denoted as following equation,

\[
T_{\text{Dis}_\text{Overall}} = 5T_{\text{UA1}_\text{SIP}} + 5T_{\text{SIP}_\text{UA2}} + 4T_{\text{SIP}} + 3T_{\text{UA}} \quad (2)
\]

Similarly, shown in Fig. 6, the total transmission time, T_{\text{Std}_\text{SIP}}, for the traditional standalone SIP server can be obtained as equation (3).

\[
T_{\text{Std}_\text{SIP}} = 9T_{\text{UA1}_\text{SIP}} + 3T_{\text{SIP}_\text{UA2}} \quad (3)
\]

The overall process delay, T_{\text{Std}_\text{Overall}}, for the traditional standalone SIP server therefore can be described as the following equation,

\[
T_{\text{Std}_\text{Overall}} = 9T_{\text{UA1}_\text{SIP}} + 3T_{\text{SIP}_\text{UA2}} + 6T_{\text{SIP}} + 4T_{\text{UA}} \quad (4)
\]

From the viewpoint of initiation procedure, we can summarize the overall process delay of discovery as well as call setup procedures in Table 2.

**Table 1 Definition of measured parameters and results**

**Table 2 Summary of overall process delay**

<table>
<thead>
<tr>
<th>Process</th>
<th>Standalone SIP server</th>
<th>Distributed pseudo SIP server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>6T_{\text{UA1}<em>\text{SIP}} + 3T</em>{\text{SIP}} + 3T_{\text{UA}}</td>
<td>2T_{\text{UA1}<em>\text{SIP}} + 2T</em>{\text{SIP}<em>\text{UA2}} + 2T</em>{\text{SIP}} + 2T_{\text{UA}}</td>
</tr>
<tr>
<td>Call</td>
<td>3T_{\text{UA1}<em>\text{SIP}} + 3T</em>{\text{SIP}<em>\text{UA2}} + 3T</em>{\text{SIP}} + T_{\text{UA}}</td>
<td>3T_{\text{UA1}<em>\text{SIP}} + 3T</em>{\text{SIP}<em>\text{UA2}} + 2T</em>{\text{SIP}} + 2T_{\text{UA}}</td>
</tr>
<tr>
<td>Setup</td>
<td>9T_{\text{UA1}<em>\text{SIP}} + 3T</em>{\text{SIP}<em>\text{UA2}} + 6T</em>{\text{SIP}} + 4T_{\text{UA}}</td>
<td>5T_{\text{UA1}<em>\text{SIP}} + 5T</em>{\text{SIP}<em>\text{UA2}} + 4T</em>{\text{SIP}} + 3T_{\text{UA}}</td>
</tr>
</tbody>
</table>

Then we deploy a test bed to measure the overall initiation delay on both traditional standalone and distributed pseudo SIP server cases. We use a sip test program for both UAs to simulate SIP initiation calls and record the message round-trip time. For the former case, the standalone SIP server is deployed in an independent Signal Board Computer (SBC) and connected via Ad-Hoc mode wireless network. For
the latter case, the pseudo SIP server is implemented and operated in each UA.

We emulate the SIP call initiation 1000 times repeatedly and continuously on each scenario. The average result for each individual parameter is listed in Table 1. The overall initiation process delay for the case of distributed pseudo SIP server is around 62 ms and for the case of traditional standalone SIP server is around 127 ms. This result indicates that the signaling process delay of VoIP system is reduced significantly by employing the distributed pseudo SIP server architecture.

If we do not take the DHCP discovery process into account for the Standalone case, equations (3) and (4) will be modified as follows, respectively.

\[ T_{Std\_SIP} = 5T_{UA1\rightarrow SIP} + 3T_{SIP\rightarrow UA2} \]  

\[ T_{Std\_Overall} = 5T_{UA1\rightarrow SIP} + 3T_{SIP\rightarrow UA2} + 4T_{SIP} + 2T_{UA} \]

The emulated results (overall delay) in this case will be around 84 ms for traditional standalone SIP server which also higher than our proposed system.

5 Conclusions

In this paper, we have proposed a distributed VoIP architecture with embedded pseudo SIP server which acts as a middleware to intermediate between SIP UA and transport layer. The proposed design has solved the bottleneck problem of standalone SIP server with massive client requests. In addition, the proposed embedded pseudo SIP server reduces the signaling delay time by integrating the registration with discovery mechanisms.

Further research in performance with different parameters is undergone. The implementation of the whole system is under developed.

Acknowledgement

The authors would like to thank the partial support from National Science Council of Republic of China, Taiwan under contracts, NSC 94-2219-E-324-001 and NSC 94-2622-E-324-009-CC3.

References:


