Prepaid Mechanism of VoIP and Messaging Services

Sok-Ian Sou, Quincy Wu, Yi-Bing Lin and Che-Hua Yeh
CSIE Dept., National Chiao Tung University, Hsinchu, Taiwan
{sisou, solomon, liny, chyei}@csie.nctu.edu.tw

***Abstract*— Session Initiation Protocol (SIP) is used for Voice over IP (VoIP) signaling and call control. Billing mechanisms for messaging and VoIP are typically deployed for postpaid services. Prepaid mechanisms for SIP-based VoIP calls toward Public Switched Telephone Network (PSTN) are seldom discussed. In this paper, we propose a prepaid mechanism that can simultaneously process messaging and VoIP calls. This prepaid mechanism is implemented in a VoIP platform developed under National Science & Technology Program for Telecommunications (NTP). Unlike previous VoIP prepaid approaches which require modifications to endpoints, our mechanism can be easily integrated to a VoIP platform without modifying existing network components.**

Index Terms — **B2BUA, NTP, Prepaid, RADIUS, VoIP**

I. Introduction

Voice over IP (VoIP) has become a major trend in telecommunications. By deploying VoIP, a service provider can offer advanced IP-based services with reduced costs. Furthermore, VoIP can be easily integrated with mobile data services to increase average revenue per user. One successful mobile data application is *Short Message Service* (SMS). In China, more than 85% of mobile data service is SMS. The number of SMS deliveries is up to 99.63 billion messages in the first half of 2004 [4], [5]. The thriving SMS market is encouraging more investments in the mobile data industry. Recently, SMS has interworked with Internet services such as mail and instant messaging [10]. By integrating SMS and VoIP in the Internet domain, these two killer applications can be further enhanced. Billing mechanisms for messaging and VoIP (especially for the VoIP calls toward the fixed-line telephones) are typically deployed for postpaid services. The prepaid mechanisms for these services are seldom studied in the literature. This paper proposes a prepaid mechanism that can simultaneously process messaging and VoIP calls. This prepaid mechanism is implemented in a VoIP platform developed under *National Science & Technology Program for Telecommunications* (NTP). This platform provides VoIP calls toward *Public Switched Telephone Network* (PSTN) and messaging service toward *Global System for Mobile communication* (GSM). We show how the prepaid mechanism can accommodate both VoIP and instant messaging through *Session Initiation Protocol* (SIP).

II. Cellular and Internet Messaging Service Interworking

SIP [12] is a standard VoIP signaling protocol defined in the *Internet Engineering Task Force* (IETF). It provides simple and efficient handling of multimedia sessions among multiple users. 3GPP adopts SIP for call control and signaling on the *IP Multimedia Subsystem* (IMS) [1]. With the growing demand of instant messaging services in Internet, RFC 3428 [3] proposes the MESSAGE method, a SIP extension that allows transfer of instant messages over the Internet. 3GPP Release 6 also supports different messaging services on the IMS [2]. With the above standard mechanisms, we have developed an Instant Message - Short Message Service (IM-SMS) Gateway to provide message exchange capability between the SIP-based network and the mobile cellular network. Based on the iSMS technology [9], the IM-SMS architecture works as follows. The IM-SMS adapter (Fig. 1(a)) receives SIP MESSAGE requests from the SIP-based VoIP network (i.e., the Internet), extracts the necessary short message information, and sends it to the iSMS gateway (Fig. 1(b)). Through the iSMS gateway, the short message is delivered to the GSM network by a GSM-compliant wireless modem (Fig. 1(c); e.g., a PCMCIA Nokia Card Phone in our implementation). This modem provides wireless access to the GSM network by sending and receiving GSM short messages just like a normal handset. The IM-SMS

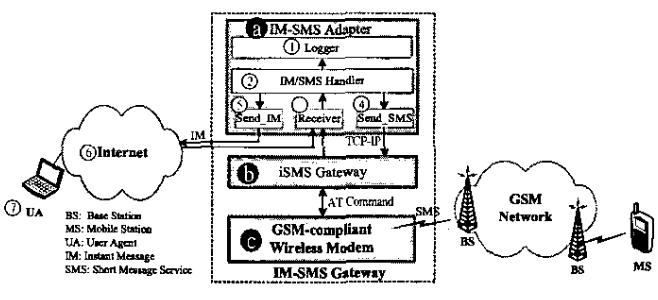

Fig. 1. The IM-SMS Gateway Architecture.

adapter is developed by utilizing jSIP library [6], a Java-based open-source library for developing SIP-based applications. With jSIP library, the IM-SMS adapter is implemented on top of the Window operating system. Fig.1 (a) shows the IM-SMS adapter architecture. Through the Receiver (Fig. 1(3)), when the IM/SMS Handler (Fig. 1(2)) receives a SIP MESSAGE request from the Internet (Fig. 1(6)), it immediately replies a response 202 Accepted to the sender indicating the request is received [3]. Suppose that the fully qualiÞed domain name of the IM-SMS Gateway is i ms ms . com. The IM/SMS Handler checks if the request contains the valid To header Þeld value such as ÒMSISDN@i ms ms . comÓ where the MSISDN (Mobile Station ISDN Number) represents the GSM phone number of the recipient. The IM/SMS Handler extracts the *MSISDN*, the sender SIP URI (specified in the *From* header field) and the SIP *message body* field, which contains the instant message. This task is implemented by the jSIP methods SipMessage.getHeaderData() and SipMessage.messageBody(). Then the IM/SMS Handler sends these extracted fields to the Send_SMS component (Fig. 1(4)) and writes a log to the Logger (Fig. 1(1)). The Send_SMS component invokes the iSMS application program Sendsm.exe [7] to create a GSM short message containing the instant message and the sender's SIP URI. The program Sendsm.exe has two parameters. The first parameter is the recipient's phone number and the second parameter is the short message. The short message is delivered to the GSM network through the wireless modem of the iSMS Gateway (Fig. 1(b)). If the short message is successfully delivered, the Send_SMS component returns a success code to the IM/SMS Handler. The IM/SMS Handler then writes a log to the Logger.

Through the IM-SMS Gateway, a GSM user can also send text messages to a SIP user in the Internet. The short message format is "IM *<recipient SIP URI>* *<message content>*". The short message recipient is the IM-SMS Gateway (i.e., the phone number of the GSM-compliant wireless modem; Fig. 1(c)). IM is the pre-defined keyword. When this keyword is encountered, the IM-SMS Gateway converts the short message to an instant message where the recipient's address is a SIP URI. Through the Receiver (Fig. 1(3)), when the IM/SMS Handler (Fig. 1(2)) receives a short message from the iSMS Gateway (Fig. 1(b)), it retrieves the *recipient SIP URI* and the *message content* fields from the short message and writes a log to the Logger. Then the IM/SMS Handler generates a SIP MESSAGE request as follows: the sender's mobile phone number (i.e., the *MSISDN*) concatenated with "`@imsms.com`" is filled in the SIP *From* header field; the *<recipient SIP URI>* part of the short message is filled in the *To* header field; the *<message content>* part is filled in the SIP *message body* field. The above task is achieved by invoking the jSIP SipCall(), SipCallMember() and SipCallMember.requestMessage() methods. Finally, the SIP MESSAGE is sent to the Internet from the Send_IM (Fig. 1(5)) component. When the recipient (i.e. SIP UA; Fig. 1(7)) receives the instant message and replies a 200 OK message, the Send_IM component informs the IM/SMS Handler. The IM/SMS Handler then writes a log to the Logger and the short message is considered delivered.

III. Prepaid Mechanism of SIP-based Services

Fig. 2 illustrates the NTP VoIP Platform architecture that consists of four major components. The *Call Server* (Fig. 2(a)) provides primary capabilities for call-session control in the NTP VoIP Platform. It processes SIP requests and responses as a SIP proxy server. It also functions as a registrar that stores the contact information of each SIP user. The *PSTN Gateway* (Fig. 2(b)) interworks the VoIP platform with the PSTN. The *SIP UA* (Fig. 2(c)) provides call functions such as dial, answer and reject. The UA can also send or receive instant messages. The IM-SMS Gateway (Fig. 2(d)) enables the SIP users sending and receiving short messages over the GSM mobile networks. The *Prepaid Mechanism* (Fig. 2(e)) controls prepaid call tolls, collects and processes the charging information. Details of the prepaid mechanism are given in this section. The IM-SMS Gateway is described in the previous section. Other components of the NTP VoIP platform are elaborated in [15]. The prepaid mechanism utilizes *Remote*

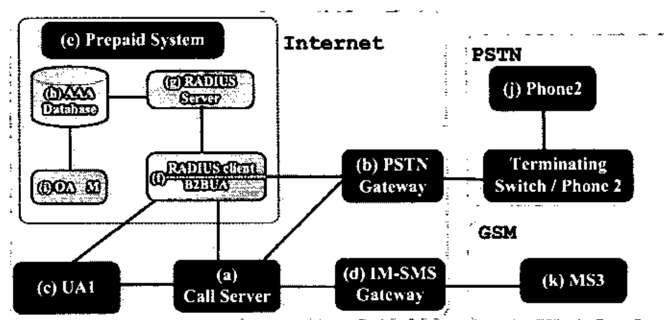

Fig. 2. The Prepaid VoIP Architecture.

Authentication Dial-In User Service (RADIUS) protocol [11] to enable centralized *Authentication, Authorization, and Accounting* (AAA) functions for NTP VoIP network access. The prepaid mechanism includes four components. The RADIUS client residing in a SIP-based *Back-to-Back UA* (B2BUA; see Fig. 2(f)) processes and exchanges SIP messages between the call parties, and terminates a prepaid call when the authorized session time for the call is expired. The RADIUS server (Fig. 2(g)) authorizes prepaid requests and replies the RADIUS client with authorization information. It also processes the RADIUS accounting messages and stores the user *Call Detail Records* (CDRs) in a non-versatile SQL-based AAA database (Fig. 2(h)). The *Operations, Administration and Maintenance* (OA&M) system (Fig. 2(i)) can be accessed from web browsers over *Secure Sockets Layer* (SSL) protocol. Through OA&M, an administrator views and/or modifies user information and browses the prepaid CDRs generated by the prepaid mechanism.

By utilizing the B2BUA technique, the prepaid function is inserted into a SIP VoIP system by breaking a SIP session into two sub-sessions. In this way, the prepaid mechanism can monitor or terminate the session when the user credit is depleted. To set up a VoIP call, the signaling message is first routed from the UA (Fig. 2(c)) to the Call Server. The Call Server can identify the charging type of the service (i.e., prepaid or postpaid) by the phone number or by the subscriber profile stored in the AAA database (Fig. 2 (h)). In the NTP VoIP platform, some telephone numbers are reserved for prepaid services. When the Call Server receives a SIP request from a prepaid phone number, it forwards the request to the B2BUA for authorization. After the authorization, the B2BUA (Fig. 2(f)) sends the authorized request back to the Call Server. The Call Server then forwards the authorized request to the called SIP devices. On the other hand, if the Call Server receives a SIP request from a non-prepaid user, it directly forwards the message to the called party without involving the prepaid mechanism. In our deployment, a prepaid phone number must have the prefix "0944021". Therefore, the prepaid functions can be easily achieved by re-configuring the call routing rules in the Call Server. An example of re-configuring routing rules in the Call Server (based on the SIP Express Router [14]) is given below:

```
if (search("From:sip:0944021[0-9][0-9][0-9]@") &&
#If this is a prepaid account
   !search("Call-ID:.*@prepaid.com")&&
   #and this request need to be authorized by the prepaid mechanism
   (method=="INVITE" || method=="CANCEL" ||
    method=="BYE" || method=="ACK" ||
    method=="MESSAGE"))
 {
          #Re-route this request to the Prepaid Mechanism
          log(1,"Prepaid-User");
          rewritehost("prepaid.com");
          forward(prepaid.com,5060);
          break;
 };
```

The above configuration assures that the request messages generated from a prepaid user are forwarded to the prepaid B2BUA for authorization. In the next subsections, we describe message flows for the prepaid call and prepaid messaging service.

A. Message flows for Prepaid Call Setup and Termination

The simplified prepaid call setup and force-termination message flows are illustrated in Fig. 3. The SIP messages are exchanged among the Call Server, the B2BUA and the PSTN Gateway. The SS7 messages are exchanged between the Terminating Switch and the PSTN Gateway [8]. The call setup message flow in Fig. 3 (Steps 1-14) is described as follows:

Step 1. UA1 sends the Call Server an INVITE message with Request-URI addressing to Phone2. According to the phone number configuration in the Call Server, UA1 is considered as a prepaid user. The Call Server forwards this message to the B2BUA for authorization.

Step 2. When the B2BUA receives the INVITE message, its residing RADIUS client sends an Access-Request message (including the prepaid phone number) to the RADIUS server.

Step 3. The RADIUS server retrieves the prepaid user's record from the AAA database. If no record is found, the RADIUS server replies with an Access-Reject message indicating that the user is not an authorized prepaid user. Otherwise, the RADIUS server replies with an Access-Accept message, which contains the available prepaid credit in the *Session-Timeout* attribute representing the maximum quota of money or time.

Step 4. When the B2BUA receives the Access-Accept message, it generates another INVITE message (for Subsession 2) to Phone2. The contact address of the B2BUA is filled in the *Contact* header field. This message is sent to the PSTN Gateway through the Call Server.

Steps 5 and 6. The PSTN Gateway generates the SS7 Initial Address message (IAM) to the Terminating Switch at the PSTN. Phone2 then starts ringing. When Phone2 is picked up, the SS7 Answer message (ANM) is sent from the Terminating Switch to the PSTN Gateway. The message indicates that the called party has answered the call.

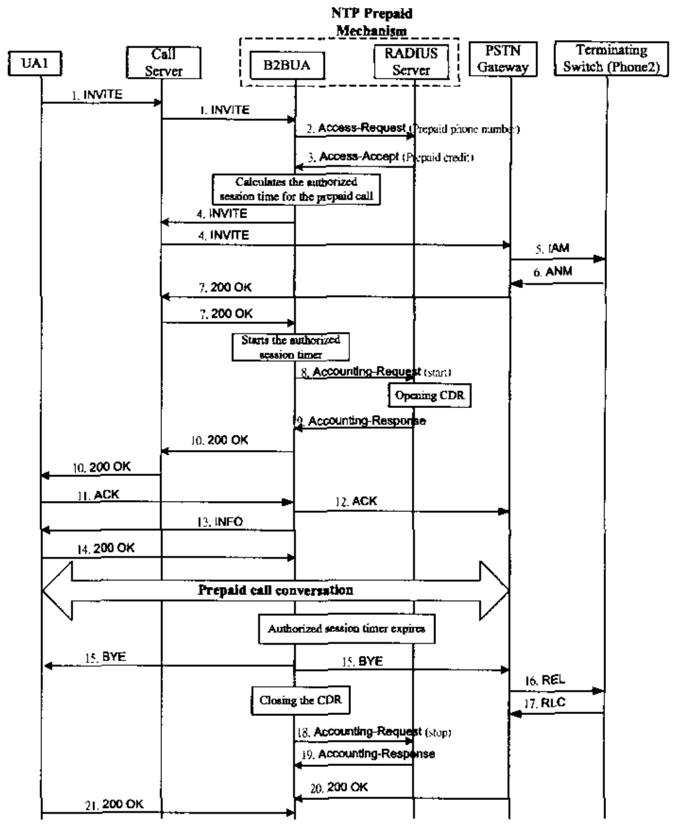

Fig. 3. Message Flows for Call Setup and Call Force-Termination.

Step 7. The PSTN Gateway generates a 200 OK message. The contact address of the PSTN Gateway is filled in the *Contact* header field. This message is routed to the Call Server and then forwarded to the B2BUA.

Steps 8 and 9. Upon receipt of the final response, the B2BUA starts an authorized session timer with the value based on the *Session-Timeout* attribute (obtained in Step 3). The RADIUS client sends an Accounting-Request message with *Status* "start" indicating the beginning of a prepaid call session. The RADIUS server creates an accounting CDR in the AAA database (via the SQL command INSERT) and acknowledges the RADIUS client with an Accounting-Response message.

Step 10. The B2BUA generates a 200 OK message to UA1. The contact address of the B2BUA is set in the *Contact* header field. This response is routed to the Call Server and then forwarded to UA1.

Steps 11 and 12. Upon receipt of the 200 OK message, UA1 sends an ACK message to the B2BUA. The B2BUA also sends an ACK message to the PSTN Gateway. At this point, the PSTN Gateway opens a UDP port for Real-time Transport Protocol (RTP) [13] so that the voice packets sent from UA1 can pass through the PSTN Gateway. The prepaid call session is established.

Step 13. The B2BUA sends an INFO message to inform UA1 about the available prepaid credit.

Step 14. When UA1 receives the INFO message, it replies with a final response 200 OK message.

When one call party terminates the prepaid call, the BYE message is sent to the B2BUA directly. Then the B2BUA triggers the Accounting-Request message with *Status* "stop" to close the CDR. If the prepaid credit is exhausted before the call is complete, the call is force-terminated by the prepaid mechanism with the following steps in Fig. 3 (Steps 15-21):

Step 15. When the authorized session timer (started in Step 8) times out, the B2BUA sends BYE messages to both UA1 and the PSTN Gateway.

Steps 16 and 17. The PSTN Gateway sends the SS7 Release (REL) message to the Terminating Switch. The Terminating Switch replies with the SS7 Release Complete (RLC) message. At this point, the PSTN Gateway closes the UDP port for the RTP connection so that subsequent voice packets sent from UA1 cannot pass through the PSTN Gateway.

Steps 18 and 19. The B2BUA sends an Accounting-Request message with *Status* "stop" and *Terminate-Cause* "Session-Timeout" to the RADIUS server indicating force termination of the prepaid call. The RADIUS Server responds with an Accounting-Response message after the accounting information (including remaining prepaid credit) is stored in the AAA database (via the SQL command UPDATE).

Steps 20 and 21. When the B2BUA receives the 200 OK messages from UA1 and the PSTN Gateway, the call is terminated.

B. Message Flow for Prepaid Messaging Service

The message flow for the prepaid messaging services reuses the message flow for VoIP. Consider the scenario where UA1 (Fig. 2(c)) sends an instant message to a GSM Mobile Station (MS3; see Fig. 2(k)) with phone number "+886936123456". Suppose that the fully qualified domain names of the prepaid mechanism and the IM-SMS Gateway are `prepaid.com` and `imsms.com`, respectively. The instant message is embedded in the message body field of a SIP MESSAGE request. The *To* header field of the SIP MESSAGE request is "+886936123456@`imsms.com`". To reuse the prepaid message flow for VoIP, the prepaid B2BUA (Fig. 2(f)) breaks the SIP session between UA1 (Fig. 2(c)) and the IM-SMS Gateway (Fig. 2(d)) into one subsession between UA1 and the B2BUA (Subsession 1) and another subsession between the B2BUA and the IM-SMS Gateway (Subsession 2). The message flow in Fig. 4 is described as follows:

Step 1. UA1 sends the Call Server a SIP MESSAGE request with *To* header field "+886936123456@`imsms.com`". The message body field of this request contains the instant message. According to the phone number configuration in the Call Server, UA1 is considered as a prepaid user, and the Call Server

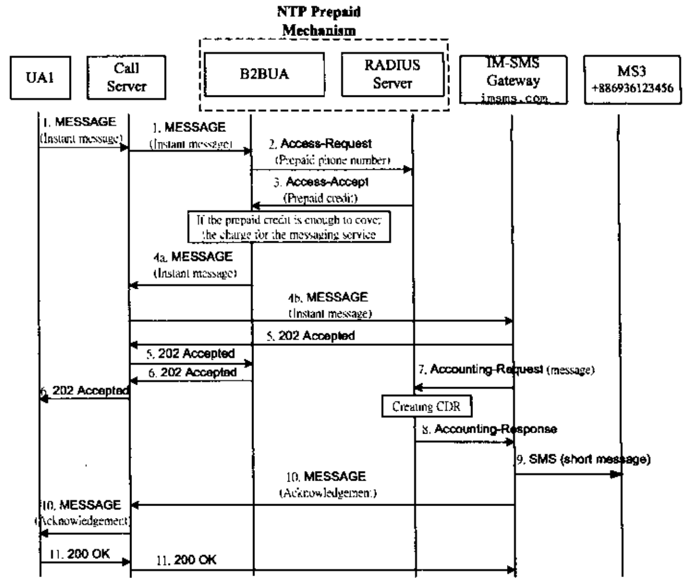

Fig. 4. Message flow for the prepaid messaging service.

forwards this request to the B2BUA for authorization and prepaid credit retrieval.

Step 2. When the B2BUA receives the MESSAGE request, its residing RADIUS client sends an Access-Request message (including the prepaid phone number) to the RADIUS server.

Step 3. The RADIUS server retrieves the UA1's record from the AAA database. If no record is found, the RADIUS server replies with an Access-Reject message indicating that UA1 is not an authorized prepaid user. Otherwise, the RADIUS server replies with an Access-Accept message. This message includes the *Session-Timeout* attribute that contains the amount of the available prepaid credit.

Step 4a. If there is enough prepaid credit to cover the charge for the messaging service, the B2BUA generates another MESSAGE request (for Subsession 2) which has a new *Call-ID* header field ended with "`@preapid.com`". The *To* header field remains the same (i.e., "+886936123456@`imsms.com`") .This new MESSAGE request containing the instant message is sent to the Call Server.

Step 4b. When the Call Server receives this authorized MESSAGE request (which has the *Call-ID* header field ended with "`@prepaid.com`"), it forwards this request to the IM-SMS Gateway.

Step 5. Upon receipt of the SIP MESSAGE request, the IM-SMS Gateway immediately replies a 202 Accepted response to the B2BUA [3]. This response indicates that the IM-SMS Gateway has received the instant message, but does not imply that the short message is already delivered to the destination.

Step 6. When the B2BUA receives 202 Accepted response from the IM-SMS Gateway, it replies 202 Accepted to UA1 through the Call Server. UA1 stops any retransmission of this MESSAGE request to the Call Server.

Steps 7 and 8. The residing RADIUS client in the B2BUA sends an Accounting-Request message with *Status* "message" to the RADIUS server. This status indicates that a prepaid messaging service is provided. Note that in RFC 2866, the RADIUS accounting protocol is designed for session-based services (with *Status* "start" and "stop" to indicate the beginning and the end of the service). We add the new *Status* "message" in the reserved value for the prepaid messaging service. The RADIUS server creates an accounting CDR in the AAA database (via the SQL command INSERT), deducts the prepaid user credit for the messaging service, and acknowledges the RADIUS client by an Accounting-Response message.

Step 9. When the IM-SMS Gateway receives the SIP MESSAGE request (containing the instant message) from the Call Server (in Step 4b), the IM-SMS Gateway converts the SIP-based instant message to a GSM short message, and then invokes the wireless modem to send this short message to MS3. To avoid illegal usage, the IM-SMS Gateway maintains an access control list consisting of authorized Call Servers. Only SIP MESSAGE requests from the authorized Call Servers are processed.

Step 10. When the short message is delivered, the IM-SMS Gateway generates a SIP MESSAGE with *message body* "*Your message is successfully delivered to +886936123456*" to UA1. This SIP MESSAGE is used as a positive acknowledgement to UA1.

Step 11. When UA1 receives this acknowledgement, it replies with the 200 OK to IM-SMS Gateway through the Call Server.

We note that the above message flow is basically the same as that for prepaid VoIP call setup. That is, the same prepaid setup flow is used for both voice call and instant messaging in our approach.

IV. Conclusions

This paper proposed a SIP-based prepaid mechanism to handle both the prepaid calls and messaging services in a VoIP system. Integration of our prepaid mechanism with the existing VoIP platform is easily achieved by re-configuring the Call Server. A prototype of our prepaid mechanism has been developed in the NTP VoIP platform connecting five major university campuses in Taiwan.

V. Acknowledgements

This work was sponsored in part by NSC Excellence project NSC93-2752-E-0090005-PAE, NSC 93-2213-E-009-100, and NTP VoIP Project under grant number NSC 92-2219-E-009-032. IIS/Academia Sinica, and ITRI/NCTU Joint Research Center.

References

[1] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; IP Multimedia Subsystem Stage 2," Technical Specification 3G TS 23.228 version 6.2.0 (2003-06), 2003.

[2] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; Multimedia messaging service Stage 1," Technical Specification 3G TS 22.340 version 6.1.0 (2003-03), 2003.

[3] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C. and Gurle, D., "Session Initiation Protocol (SIP) Extension for Instant Messaging," IETF RFC 3428, December, 2002.

[4] Chinanex, "http://www.chinanex.com/insight/mob0804.htm/", 2004.

[5] Interfax, "http://www.interfax.com/com?id=5757947&item=Chin", 2004.

[6] jSIP Project, "http://jsip.sourceforge.net/".

[7] Lin, P., Rao, H. C.-H., and Lin, Y.-B. et al., "Short Message Service Programming for Wireless Internet," Chinese Version, Wei Keg Publishing Co. Taiwan, 2002.

[8] Lin, Y.-B., and Chlamtac, I., "Wireless and Mobile Network Architectures," John Wiley & Sons, 2001.

[9] Rao, C.-H., Chang, D.-F., and Lin, Y.-B., "iSMS: An Integration Platform for Short Message Service and IP Networks," *IEEE Network*, vol. 15, no. 2, pp. 48-55, 2001.

[10] Rao, H. C.-H., Cheng, Y.-H., Chang, K.-S., and Lin, Y.-B., "iMail: A WAP Mail Retrieving System," *Information Sciences*, vol. 151, pp. 71-91, 2003.

[11] Rigney, C., Willens, S., Rubens, A. and Simpson W., "Remote Authentication Dial In User Service (RADIUS)," IETF RFC 2865, June, 2000.

[12] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E., "SIP: Session Initiation Protocol," IETF RFC 3261, June, 2002.

[13] Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 1889, January, 1996.

[14] SER, "http://www.iptel.org/ser/".

[15] Wu, Q., Chen, W.-E., Pang, A.-C., Lin, Y.-B. and Chlamtac, I., "NTP VoIP Testbed: A SIP-based Wireless VoIP Platform," To appear as a book chapter in *Handbook of Algorithms for Mobile and Wireless Networking and Computing* by CRC Press.