VoIP   3G Call Agent

93   2   5

NTP VoIP Testbed: A SIP-based Wireless VoIP Platform

Quincy Wu, Whai-En Chen
Department of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, Taiwan

Ai-Chun Pang
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan

Yi-Bing Lin
Department of Computer Science and Information Management
Providence University
Taichung, Taiwan
liny@csie.nctu.edu.tw

Imrich Chlamtac
Center for Advanced Telecommunications Systems and Services
The University of Texas at Dallas
Richardson, TX 75083-0688
chlamtac@utdallas.edu

1 Introduction

Over a century, telephony industry was mainly established based on circuit-switched technologies. In the recent years, *Internet Protocol* (IP) technology has been utilized to transport voice over packet-switched network [15]. This approach consumes lower bandwidth, and therefore reduces communications costs. For example, a typical coding algorithm used by circuit-switched telephony is based on *International Telecommunications Union* (ITU) Recommendation G.711, which requires 64Kbps bandwidth in each direction for a telephone call [14]. On the other hand, by

1

adopting silence suppression technique and advanced coding algorithms (such as those described in ITU G.729), voice traffic can be transported over IP network at 8Kbps [7], with approximately the same voice quality as circuit-switched voice. Many Internet telephony service providers follow this approach to offer cost effective voice services. Since IP is the de facto standard for data transactions, VoIP is advantageous in integration of voice and data [6]. One of the most popular protocols for VoIP signaling and call control is *Session Initiation Protocol* (SIP) that supports functions to facilitate integration of user presence, instant messaging, and multimedia communications [23]. Specifically, SIP is chosen as the signaling protocol in IP Multimedia Subsystem in 3GPP specifications [17].

Under the *National Telecommunication Development Program* (NTP), we have established a VoIP testbed that allows deployment of SIP-based wireless VoIP applications. This paper presents the operations and the numbering plan of the NTP VoIP testbed, and the wireless VoIP applications developed in this testbed. We first introduce SIP. Then we discuss the design philosophy of call servers in the NTP VoIP testbed. We propose an approach to support real-time VoIP through *General Packet Radio Service* (GPRS) that interworks with the NTP VoIP testbed. We also describe a *WLAN-based GPRS Support Node* approach for accessing VoIP services.

2 NTP VoIP Testbed: Architecture and Protocol

Figure 1 illustrates the NTP VoIP testbed architecture. Components in this VoIP system interact with each other through the SIP protocol. These components are described as follows.

Call server (Figure 1(a)) provides primary capabilities for call-session control in the NTP VoIP testbed. A call server 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. Details of proxy and registrar servers will be given in Section 2.1. At the current stage, the NTP call server is implemented on Window 2000 server running on an industrial PC. Figure 2 illustrates the operations, administration and maintenance (OAM) system of the call server. This figure shows the SIP phone management page of the call server.

PSTN Gateway (Figure 1(b)) supports interworking between the NTP VoIP testbed and the *Public Switched Telephone Network* (PSTN), which allows IP phone users to reach other PSTN
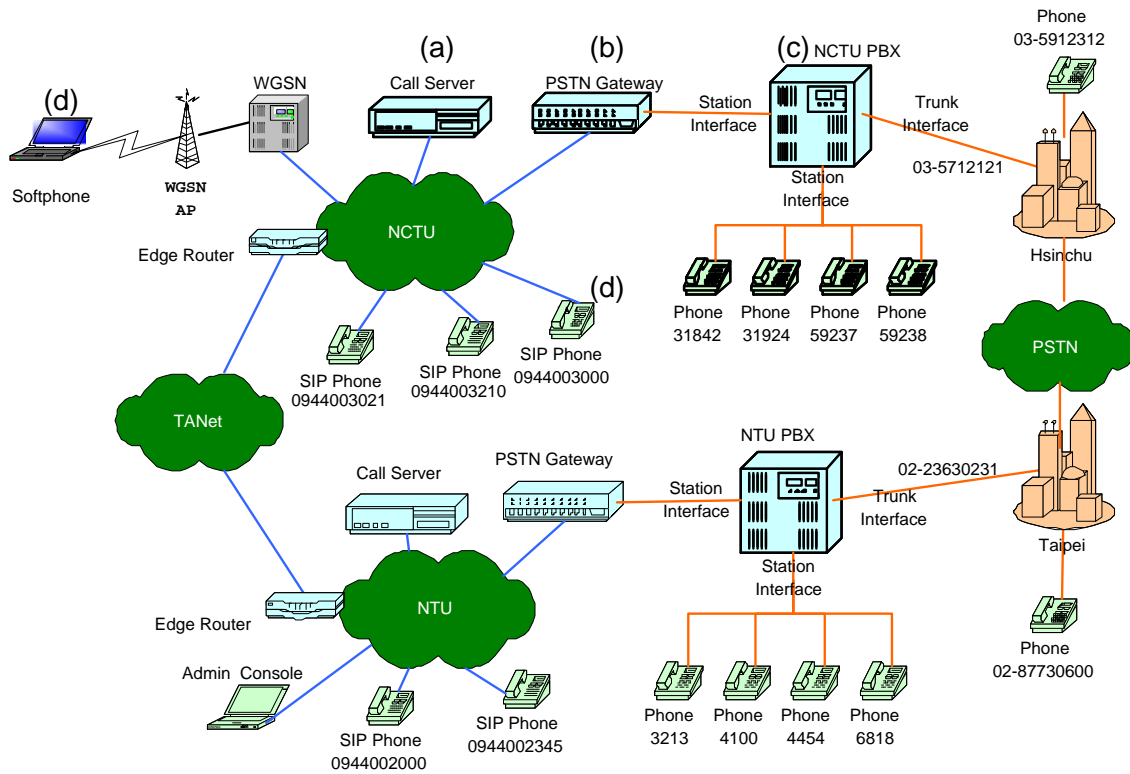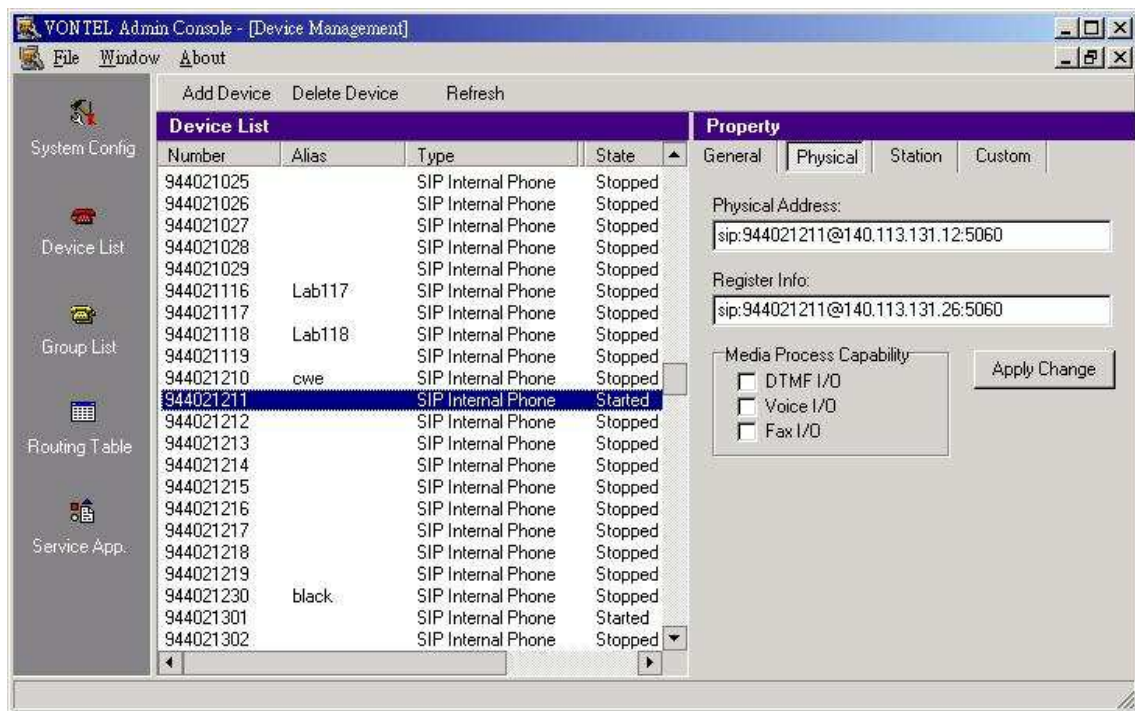
Figure 1: NTP VoIP Testbed

Figure 2: OAM of the NTP Call Server

(a) ITRI PSTN Gateway


(b) Cisco 2600

Figure 3: NTP PSTN Gateways

users directly or indirectly through *Private Branch Exchange* (PBX; see Figure 1(c)). Two types of PSTN gateways have been deployed in the NTP VoIP testbed, including the gateway developed by Industrial Technology Research Institute (ITRI; see Figure 3 (a)) and Cisco 2600 (Figure 3 (b)).

SIP User Agent (UA) is a hardware-based or a software-based SIP phone client (Figure 1(d)) that provides basic call functions such as dial, answer, reject, hold/unhold, and call transfer. In the NTP VoIP testbed, we have installed the SIP UA in software-based terminals including desk-top computers, notebooks (with or without WLAN access), and PDAs (with WLAN access only). The GUI for softphone is shown in Figure 4. We also include hardware-based SIP phones manufactured by Cisco, Leadtek, Pingtel, and Snom (see Figure 5).

At the time of writing, the NTP VoIP testbed has been deployed in four sites: National Taiwan University (NTU), National Tsing Hua University (NTHU), National Chiao Tung University (NCTU) and ITRI. NTU is in Taipei City while the other three organizations are in Hsinchu City. Figure 1 illustrates the NCTU and NTU sites of the NTP VoIP testbed. A call server and a PSTN gateway are deployed in each site, in accompany with several SIP hardphones and software-based SIP

4

(a) SIP UA Developed by NTP/ITRI
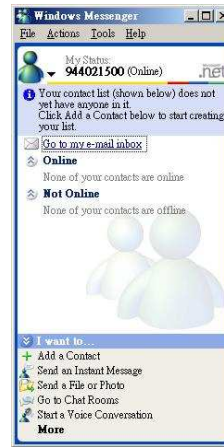
(b) Windows Messenger 4.7 (including a SIP UA) on Windows XP)

Figure 4: GUI of NTP VoIP Softphone

(a) Cisco Hardphone

(b) Leadtek Hardphone

(c) Pingtel Hardphone

(d) Snom Hardphone

Figure 5: NTP User Terminals

phone clients.

Depending on the locations of the calling party and the called party, and whether they are SIP phones or PBX/PSTN phones, there are four call setup scenarios in the NTP VoIP testbed.

Scenario 1. If a SIP phone UA1 in NCTU attempts to call another SIP phone UA2 on the same campus, the call setup signaling messages are delivered between UA1 and UA2 indirectly through the NCTU call server. The voice connection is established directly between UA1 and UA2 without involving the NCTU call server.

Scenario 2. If a SIP phone UA1 in NCTU attempts to call another SIP phone UA2 in NTU, UA1 first signals to the NCTU call server. From the destination address, the NCTU call server determines that UA2 is located on the NTU campus, and forwards the call setup request to the NTU call server. The NTU call server retrieves the registry information of the called party and set up the call to UA2. Again, the voice connection is established directly between UA1 and UA2 without involving the NCTU call serve.

Scenario 3. If a SIP phone UA1 in NCTU makes a call to a PBX phone P1 in NCTU or a traditional phone in the PSTN, the call setup procedure is similar to that of Scenario 1 with the following exceptions. The NCTU call server determines that the called party is not a SIP phone and routes the call to the NCTU PSTN gateway. The PSTN gateway then sets up the call to P1 directly (if P1 is a PBX phone) or to the PSTN central office for further processing.

Scenario 4. If a SIP phone UA1 in NCTU attempts to call a PBX phone P1 in NTU, then the NCTU call server will forward this request to the NTU call server. Similar to Scenario 3, the NTU call server will route this call to the NTU PSTN gateway for further processing.

In order to reach a SIP phone from the PSTN, this SIP phone must be assigned a telephone number. Following Taiwan's numbering plan (based on E.164 recommendation [5]), the telephone number for a SIP phone in the NTP VoIP testbed is of the format `0944-nnn-xxx`. The first four digits 0944 are the service code for NTP VoIP testbed approved by DGT of Taiwan for experimental usage. In Japan *Electronic Number* (ENUM) trial, the prefix 050 is used for VoIP. In Taiwan, the code 070 will be reserved for VoIP services. In the future, the 0944 code will be replaced by 070. The next three digits `nnn` refers to the site number. In the current configuration, code 001 represents the NTHU cite, code 002 represents the NTU cite, and code 003 represents the NCTU

6

cite. The remaining digits xxx represent a customer line number automatically generated by the NTP VoIP registration system. Some numbers are preserved for emergency call or special services. To simplify our discussion, in the remainder of this chapter, the IP phones will be identified by SIP URI (to be elaborated) instead of telephone number mentioned above.

In this section, we describe the SIP protocol and call flows. Then we elaborate on the call server developed in the NTP VoIP testbed.

2.1 SIP Signal Protocol

Session Initiation Protocol (SIP) [21] is a signaling protocol for creating IP multimedia sessions (e.g., voice and video streaming). Following a text-based HTTP-like format, SIP conjuncts with protocols such as *Session Description Protocol* (SDP) [12] and *Real-time Transport Protocol* (RTP) [22]. SDP describes multimedia information of a session, including media type, IP address, transport port, and codec. RTP transports real-time multimedia data such as voice and video data. SIP supports the following features:

User location determines the end terminal location for communication;

User availability determines the willingness of the called party to engage in communications;

User capability determines the media and media parameters to be used;

Session setup: establishes session parameters at both called and calling parties;

Session management: transfers and terminates of sessions, modifies session parameters, and invokes services.

To handle setup, modification, and teardown of a multimedia session, SIP messages are exchanged between the call parties, including the SIP *requests* from the calling party to the called party, and SIP *responses* from the called party to the calling party. IETF RFC 3261 [21] defines six types of requests that are utilized in the NTP VoIP testbed.

1. REGISTER is sent from a SIP UA to the call server for registering contact information.

2. INVITE is sent from a calling UA to a called UA to initiate a session for, e.g., audio, video, or a game.

3. `ACK` is sent from the calling UA to the called UA to confirm that the final response has been received.

4. `CANCEL` is sent from the calling UA to the called UA to terminate a pending request.

5. `BYE` is issued by either the calling or the called UAs to terminate sessions.

6. `OPTIONS` is sent from a SIP UA to query the capabilities of another UA.

All SIP requests are acknowledged by SIP responses. A SIP response consists of a numeric status code and the associated textual phrase. Examples of the codes are `100 (Trying)`, `180 (Ringing)`, `200 (OK)`, `302 (Moved Temporarily)`, and `487 (Request Terminated)`.

Every SIP entity is identified by a unique identification called SIP *Uniform Resource Identifier* (URI). The SIP URI is of the format `sip:username@hostname:port`. In this format, the prefix `sip:` indicates that the whole string is a SIP URI. The `hostname` is the name of the host for the SIP phone and `username` is a local identifier representing the SIP phone on the host. The `port` is the transport port to receive the SIP message on `hostname`, which has a typical value "5060". A SIP URI example is `sip:george@work.com:5060`.

2.2   Basic SIP Call Flow

As its name implies, the primary function of SIP is session initiation (call setup). The basic SIP call setup between two UAs, i.e., UA1 (the calling party) and UA2 (the called party), is illustrated in Figure 6. The procedure is described as follows.

Step A.1. UA1 sends the `INVITE` request to UA2. The `INVITE` message includes the SIP URI of UA2 and SDP message that describes the RTP information of UA1 (including the IP address and port number). This RTP information will be utilized by UA2 to send VoIP packets to UA1.

Step A.2. Upon receipt of the `INVITE` request, UA2 replies the `100 Trying` response. This message is a *provisional response* to inform UA1 that the `INVITE` request is in progress at UA2.
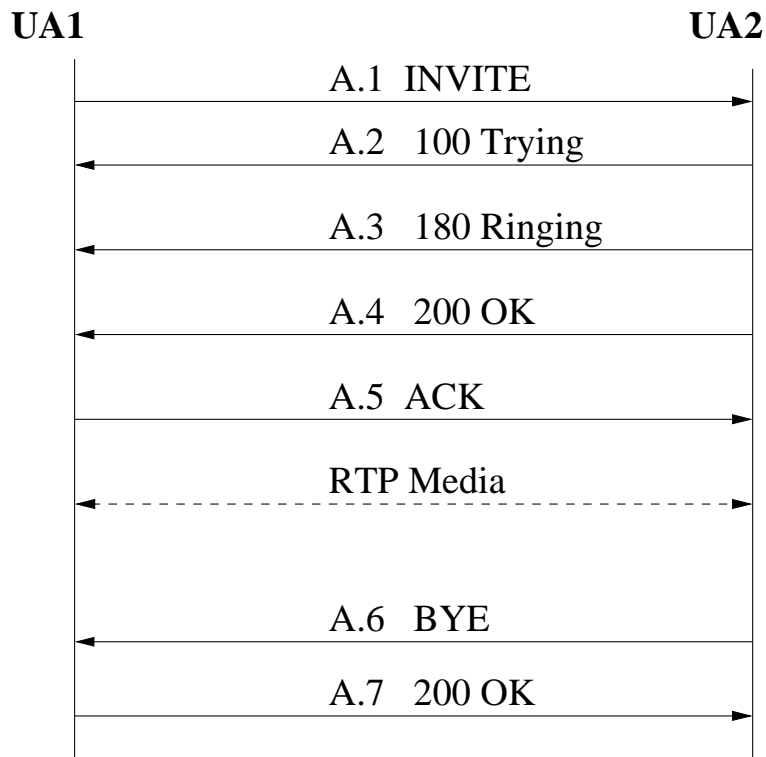
8

Figure 6: Basic SIP Call Setup Procedure

Step A.3. UA2 plays an audio ringing tone to alarm the called user that an incoming call arrives. UA2 sends the `180 RINGING` response to UA1. This message is a provisional response that asks UA1 to play an audio ringback tone to the calling user.

Step A.4. Once the called user picks up the handset, UA2 sends a `200 OK` response to UA1, which is a final response. The `200 OK` response includes the SDP message that describes the RTP information of UA2 (i.e., IP address and the transport port).

Step A.5. Upon receipt of the `200 OK` response, UA1 sends the `ACK` request to UA2 as an acknowledgement. At this point, the session is established and the conversation begins; i.e., UA2 starts sending the RTP packets to UA1 based on the SDP parameters obtained at Step A.1. Similarly, UA1 sends the RTP packets to UA2 according to the SDP parameters obtained in Step A.4.

Step A.6. Either one of the call parties can terminate the call. Assume that UA2 terminates this session by sending the `BYE` request to UA1.

Step A.7. Upon receipt of the `BYE` request, UA1 replies the `200 OK` response to confirm that the

call has been terminated.

An example of the `INVITE` request sent in Step A.1 is shown below:

```
INVITE sip:mary@pc3.home.net SIP/2.0
Via:  SIP/2.0/UDP station5.work.com
From:  George <sip:george@station5.work.com>
To:  Mary <sip:mary@pc3.home.net>
Call-ID: 123456@station5.work.com
CSeq:  1 INVITE
Content-Length:  421
Content-Type:  application/sdp

v=0
c=IN IP4 140.113.131.23
m=audio 9000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

In this example, the `From`, `To`, and `Call-ID` header fields in a SIP message uniquely identify the session. These header fields are never modified during the session, except that some tags may be added to the `To`/`From` header fields. The `From` header field indicates that this call is initiated from `sip:george@station5.work.com`. In this example, SIP allows a display name to be used with the SIP URI. When the called party is alerted, his/her SIP terminal will display the name `George` instead of the SIP URI `sip:george@station5.work.com`.

The `Content-Type` header field indicates that the message body contained in this SIP request is described by SDP. The SDP message provides necessary information to establish a media session.

2.3 NTP VoIP Call Server

In addition to user agent , SIP defines three types of servers: proxy server, redirect server and registrar.

A SIP *proxy server* accepts the SIP requests from a UA, and forwards them to other servers or UAs, perhaps after performing some translations. For example, to resolve the SIP URI in the `INVITE` request, the proxy server consults the *location service* database to obtain the current IP address and transport port of the called UA. After location information retrieval, the proxy server forwards the `INVITE` message to the called UA.

A *redirect server* accepts a request from a calling UA, and returns a new contact address to the calling UA. Similar to the proxy server, the redirect server may query the location service database to obtain the called party's contact information. Unlike the proxy server, the redirect server does not forward the `INVITE` message. Instead, it returns the contact information to the calling UA.

The basic call setup procedure described in Section 2.2 assumes that the calling party knows the physical location of the called party. In the real world, a SIP user may change the location, and it may not be possible for the calling party to know the exact location of the called party. To solve this problem, SIP provides a registration network entity called *registrar* to locate the moving users. A SIP registrar only accepts the `REGISTER` messages. A UA can periodically register its SIP URI and contact information (specifically, the IP address and the transport port) to the registrar. The registrar stores the contact information in the location service database. Therefore, when a UA moves around different networks, the registrar always maintains the actual contact information of the UA. SIP `REGISTER` request is similar to the `RegistrationRequest (RRQ)` message sent from a terminal to a gatekeeper in H.323 [13]. In most implementations, the location service database, the SIP registrar and the proxy server are integrated together. In NTP VoIP testbed, these server functions are implemented in a call server.

Figure 7 shows the message flow for SIP registration in the NTP VoIP testbed. When a UA logs in at host `station5.work.com`, a `REGISTER` request is sent to the call server (the registrar function). In this message, the `Via` header field contains the network nodes visited by the request so far. The `From` header field indicates the address of the individual who initiates the registration request. The `To` header field indicates the "target user" being registered. The `Contact` header field contains the contact address of the user, at which the user can be reached. The call server (the proxy function) retrieves the contact address from the location server database by using the `To` header field as the searching key. Generally, the `From` and `To` fields are identical when a UA registers for itself. In some exceptions, these two fields may have different values. For example, a secretary Mary may use her SIP UA to register the location of the UA for her boss George. In this case, the `From` header field is the SIP URI of Mary's UA, while the `To` header field is the SIP URI of George's UA. Mary may set the `Contact` header field to the SIP phone on her desk, so that she can answer the calls when people call George. She may also set the `Contact` address to a voicemail server. In this case, all SIP calls to George are immediately forwarded to the voicemail system.

UA1   **NTP VoIP Call Server**
                             (Registrar Function)

REGISTER sip:register.work.com SIP/2.0
Via: SIP/2.0/UDP station5.work.com
From George <sip:george@work.com>
To: George <sip:george@work.com>
Call−ID: 123408@station5.work.com
CSeq: 31842 REGISTER
Contact: George <sip:george@station5.work.com:5060>
Content−Length: 0

SIP/2.0 200 OK
Via: SIP/2.0/UDP station5.work.com
From George <sip:george@work.com>
To: George <sip:george@work.com>
Call−ID: 123408@station5.work.com
CSeq: 31842 REGISTER
Contact: George <sip:george@lab3.work.com:5060>;
         expires=2305
Contact: George <sip:george@station5.work.com:5060>;
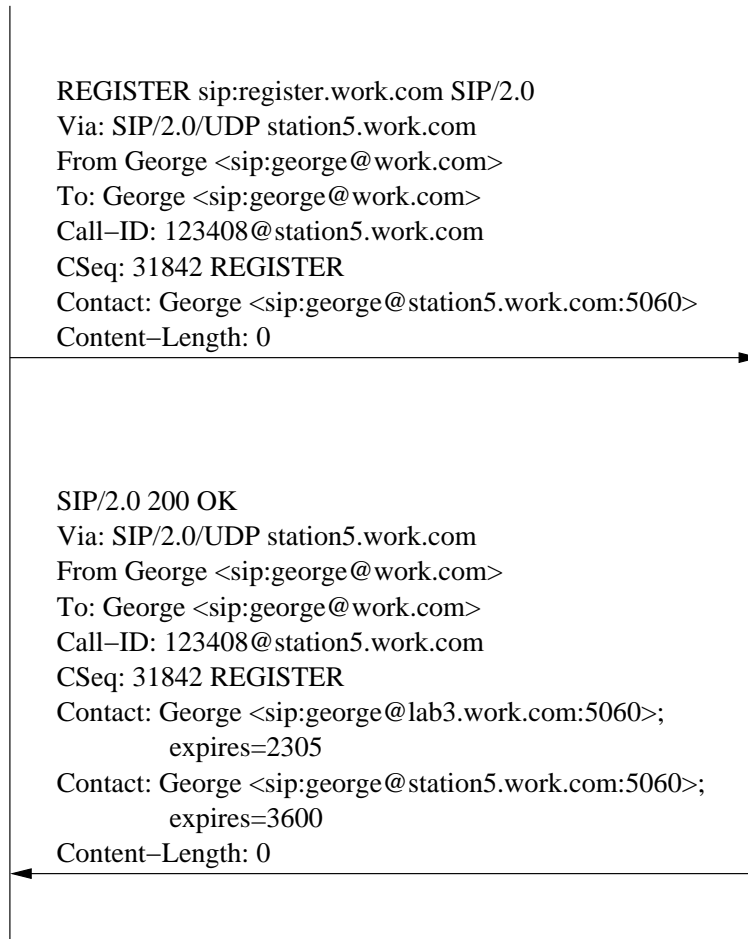   expires=3600
Content−Length: 0
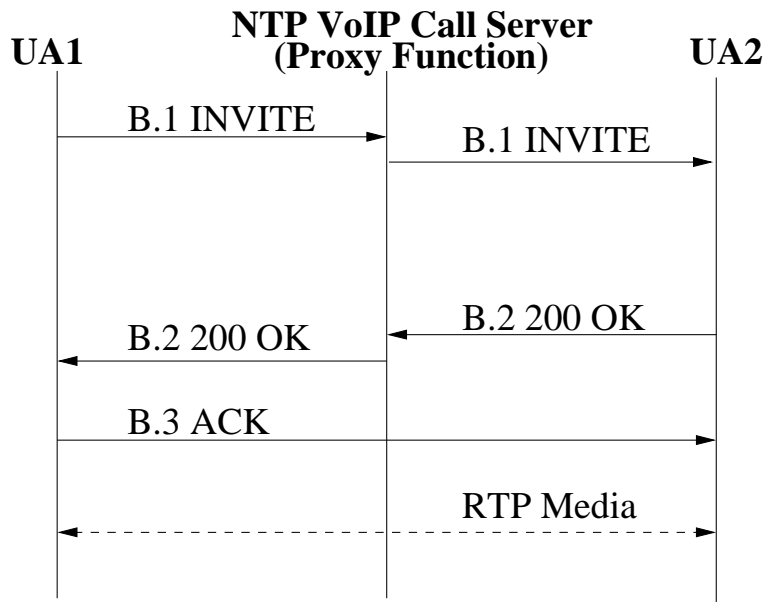
Figure 7: SIP Registration Procedure

Figure 8: Call Setup via the NTP VoIP Call Server

If a UA has registered to the call server, then other UAs do not need to know the exact location of the UA when they attempt to make calls to this UA. All they need to do is sending the `INVITE` request to the call server (the proxy function). The call server will look up the location service database to find out the exact contact address of the called party. By introducing the registration mechanism, SIP supports personal mobility where a UA only needs to maintain one externally visible identifier regardless of its network location. The call flow involving the call server is illustrated in Figure 8 with the following steps.

Step B.1. UA1 attempts to make a call to UA2. UA1 first sends the `INVITE` request to the NTP VoIP call server (the proxy function). The call server may query the location service database to obtain the contact information of UA2. Then it forwards the `INVITE` message to UA2.

Step B.2. If the incoming call is accepted, UA2 replies the `200 OK` message to UA1 through the call server (for simplicity, the provisional responses illustrated in Figure 6 are not shown here).

Step B.3. Upon receipt of the `200 OK` response, UA1 sends the `ACK` message directly to the UA2 without involving the call server. The conversation starts.

In the registration example illustrated in Figure 7, the `200 OK` response contains more than one `Contact` header fields. This SIP feature implies that a user is allowed to register for more than

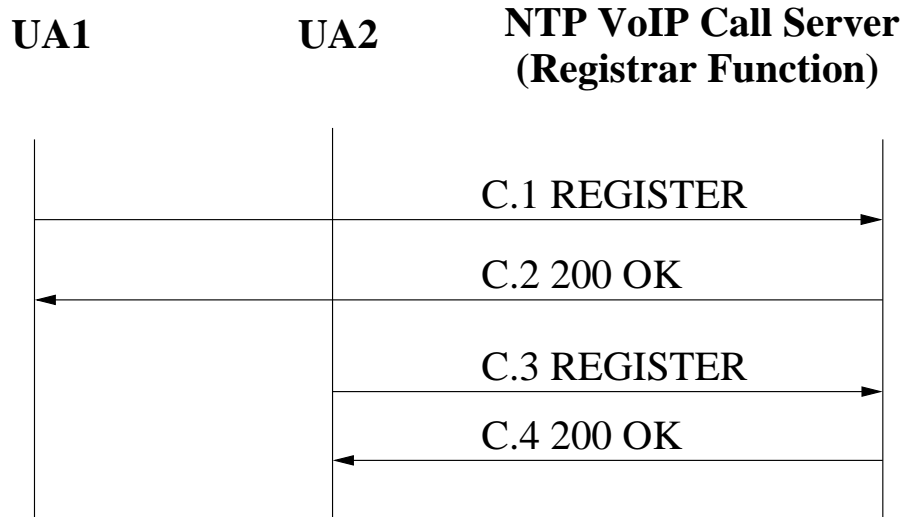**UA1**   **UA2**   **NTP VoIP Call Server**
(Registrar Function)

Figure 9: SIP Forking: Registration

one UAs. If multiple contacts are given in registration, then during call setup, the NTP VoIP call server will send the `INVITE` message to a number of locations at the same time. This operation is known as *forking*.

The forking mechanism provides an interesting feature in NTP VoIP testbed. Suppose that George has two SIP phones sharing a public address `sip:george@phone.com`, one at home and one at his office. Suppose that the two SIP phones UA1 and UA2 of George share the above public address with the contact addresses `sip:george@home.com` and `sip:george@work.com`, respectively. When someone dials George's phone number (his public address), both phones ring. When George picks up the nearest phone, the other phone stops ringing. Figure 9 illustrates how this feature is enabled in the NTP VoIP testbed:

Step C.1 When UA1 is turned on, it sends the `REGISTER` request to the NTP VoIP call server (the registrar function).

Step C.2 The call server binds the contact address of UA1 `sip:george@home.com` with George's public address `sip:george@phone.com`.

Step C.3 UA2 is turned on and sends the `REGISTER` request to the call server.

Step C.4 Similarly, the contact address of UA2 `sip:george@work.com` is bound with George's public address `sip:george@phone.com`.
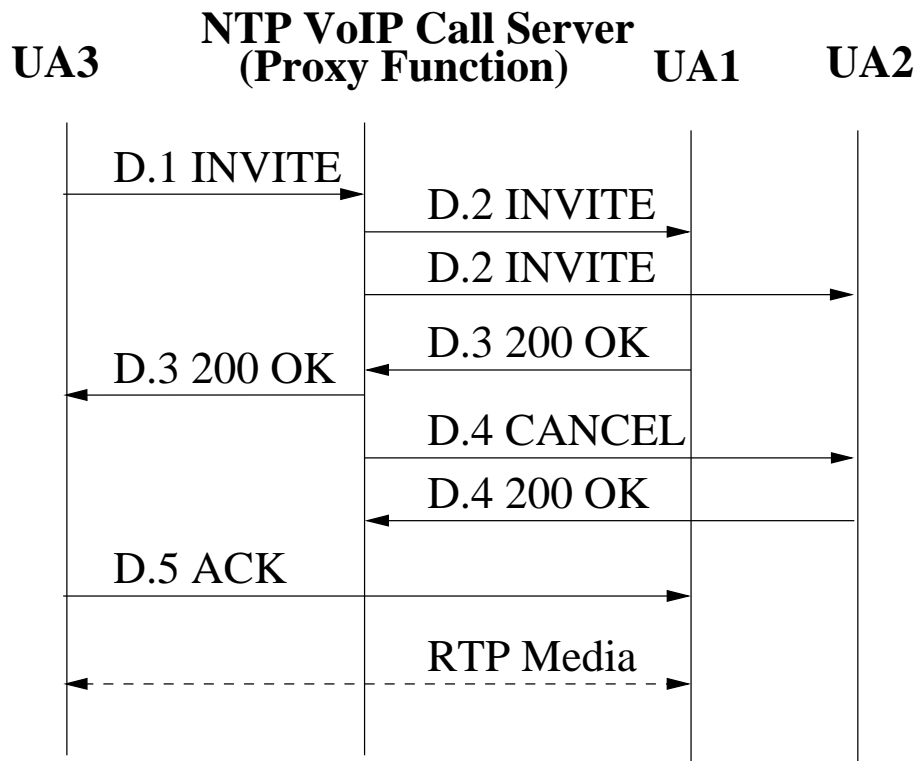
14

Figure 10: SIP Forking: Call Setup

At this point, both contact addresses `sip:george@home.com` and `sip:george@work.com` are associated with George's public address in the call server. When Mary makes a VoIP call to George, the following steps are executed (see Figure 10).

Step D.1 Mary's IP phone UA3 sends the `INVITE` request to the NTP VoIP call server (the proxy function). This message specifies George's public address `sip:george@phone.com`. The call server queries the location service database to retrieve the contact addresses of George; that is, `sip:george@home.com` and `sip:george@work.com`.

Step D.2 The call server forwards the `INVITE` request to UA1 and UA2 simultaneously. Both phones ring at the same time.

Step D.3 Suppose that George answers UA1. Then this SIP phone will send the `200 OK` message to the call server. The call server forwards this response to UA3.

Step D.4 The call server sends the `CANCEL` request to UA2. UA2 stops ringing and sends the `200 OK` response to the call server.
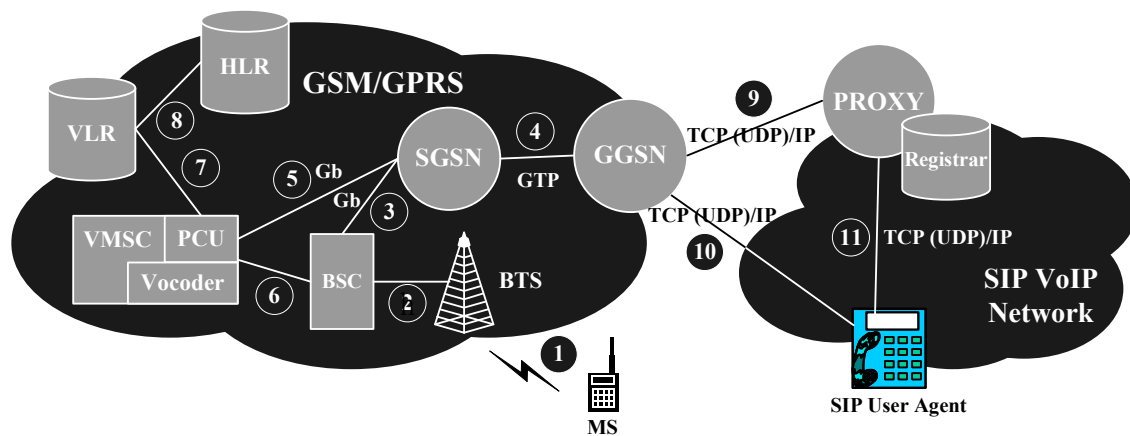
Figure 11: GSM/GPRS Architecture for VoIP

Step D.5 UA3 replies ACK to UA1 directly, and the RTP session is established between UA3 and UA1.

Forking is useful in combining various services. For example, this feature has been exploited to build a voice mail service in the NTP VoIP testbed.

3 Wireless VoIP Service

Several approaches have been proposed to integrate VoIP services with wireless technologies [20, 19]. This section describes wireless VoIP services that can be implemented in NTP VoIP testbed. We consider two types of wireless networks that have been interworked with NTP VoIP testbed: *General Packet Data Service* (GPRS) network [16] and *WLAN-based GPRS Support Node* (WGSN) network [11].

3.1 VoIP Mobile Switching Center Approach

Figure 11 shows an integrated system for VoIP services over GSM/GPRS. In this figure, the GPRS network consists of *Serving GPRS Support Node* (SGSN) and *Gateway GPRS Support Node* (GGSN) [2]. The GGSN provides interworking with external data networks (e.g., the NTP VoIP testbed), and is connected with SGSNs via an IP-based GPRS backbone network. The SGSN connects the GPRS core network to GSM radio network. Both GGSN and SGSN communicate with

the the mobility databases including the *Home Location Register* (HLR) and the *Visitor Location Register* (VLR), to track the locations of mobile subscribers.

GPRS reuses the existing GSM infrastructure to provide end-to-end packet-switched services. In the GSM radio network, a *Mobile Station* (MS) communicates with a *Base Station Subsystem* (BSS). A BSS consists of several *Base Transceiver Stations* (BTSs) and a *Base Station Controller* (BSC). A BTS contains transmitters, receivers and signaling equipment specific to the radio interface, which provides the MSs radio access to the GSM/GPRS network. The BSC supports radio channel allocation/release and the handoff management. In the circuit-switched service domain, the *Mobile Switching Center* (MSC) connects to the BSS through one or more E1/T1 lines, which performs the switching functions for mobile applications. In the packet-switched service domain, a *Packet Control Unit* (PCU) is implemented in the BSC to connect to an SGSN. The BSC forwards circuit-switched calls to the MSC, and packet-switched data (through the PCU) to the SGSN.

By introducing a new core network node called *VoIP MSC* (VMSC), we propose real-time VoIP services over GPRS without modifying the existing GSM/GPRS network nodes. A VMSC is a router-based softswitch, which communicates with BSCs, VLR, HLR, and MSCs (either GSM MSCs or VMSCs) just like a standard MSC. The VMSC also interfaces with the PSTN via the SS7 *ISDN User Part* (ISUP) protocol [16]. Unlike MSC, a VMSC communicates with an SGSN through the GPRS Gb interface (based on frame relay). In other words, while an MSC delivers voice traffic through circuit-switched trunks, a VMSC delivers VoIP packets through the IP-based GPRS core network. In Figure 11, the data path of a GPRS MS is (1)↔(2)↔(3)↔(4). The voice path in the VMSC approach is (1)↔(2)↔(6)↔(5)↔(4). In this voice path, (1)↔(2)↔(6) is circuit switched as in GSM, and (5)↔(4) is packet switched in GPRS. By using the circuit-switched based GSM air interface, the VMSC approach provides better QoS for real-time communications. Both standard GSM MSs and GPRS MSs can set up calls in the VMSC-based network as if they are in the standard GSM/GPRS network. The VMSC is anticipated to be cheaper than an MSC in terms of the implementation costs.

At a VMSC, the voice information is translated into GPRS (i.e., SIP) packets through vocoder and PCU. Then the packets are delivered to the GPRS network through the SGSN (see path (5)↔(4)). An IP address is associated with every MS attached to the VMSC. In GPRS, the IP address can be either statically or dynamically created for a GPRS MS. Creation of the IP address is performed by the VMSC through the standard GPRS *Packet Data Protocol* (PDP) context activation proce-

17

dures [9]. The VMSC maintains an MS table. The table stores the MS *Mobility Management* (MM) and PDP contexts such as *Temporary Mobile Subscriber Identity* (TMSI), *International Mobile Subscriber Identity* (IMSI), and the QoS profile requested. These contexts are the same as those stored in a GPRS MS (see Section 13.4, GSM 03.60 [9]). In the VMSC approach, SIP is used to support voice applications, where the VMSC serves as the SIP UA for any MS engaged in a VoIP session.

In Figure 11 the call server in NTP VoIP testbed performs standard functions for SIP proxy server and registrar (such as request forwarding and user location recording). In a connection between a SIP user agent and a GSM MS, the TCP(UDP)/IP protocols are exercised in links (9), (10) and (11). The GPRS tunneling protocol [10] is exercised in link (4), and the GPRS Gb protocol [8] is exercised in links (3) and (5). Standard GSM protocols are exercised in links (1), (2), (6), (7) and (8). The SIP protocol is implemented on top of TCP(UDP)/IP, and is exercised between the VMSC and the nodes in the NTP VoIP testbed. The SIP packets are encapsulated and delivered in the GPRS network through the GPRS tunneling protocol.

In GSM/GPRS, a registration procedure is performed when an MS is turned on or when the MS moves to a new location (referred to as a routing area in GPRS [16, 18]). In Figure 11, the GSM registration messages are delivered through the path $(1)\leftrightarrow (2)\leftrightarrow (6)\leftrightarrow (7)\leftrightarrow (8)$. Then the PDP context for this MS is activated, and the SIP messages (i.e., `REGISTER` and `200 OK`) are delivered through the path $(1)\leftrightarrow (2)\leftrightarrow (6)\leftrightarrow (5)\leftrightarrow (4)\leftrightarrow (9)$. Details of the SIP registration is described in Figure 7 of Section 2.3. When outgoing/incoming calls arrive at the MS, the call origination/termination procedures are performed. The signaling path for GSM call setup and SIP `INVITE` is similar to that for the registration procedure. After the call is established, the voice path is $(1)\leftrightarrow (2)\leftrightarrow (6)\leftrightarrow (5)\leftrightarrow (4)\leftrightarrow (10)$ in Figure 11. Details of SIP call setup procedure is given in Figure 8 of Section 2.3.

3.2 WLAN-based GPRS Support Node

This subsection describes the architecture and features of the *WLAN-based GPRS Support Node* (WGSN) jointly developed by NCTU and *Computer and Communications Laboratories* (CCL) of ITRI [11]. WGSN provides the functions of SIP *Application Level Gateway* (ALG) and *Authentication/Authorization/Accounting* (AAA) in the NTP VoIP testbed.
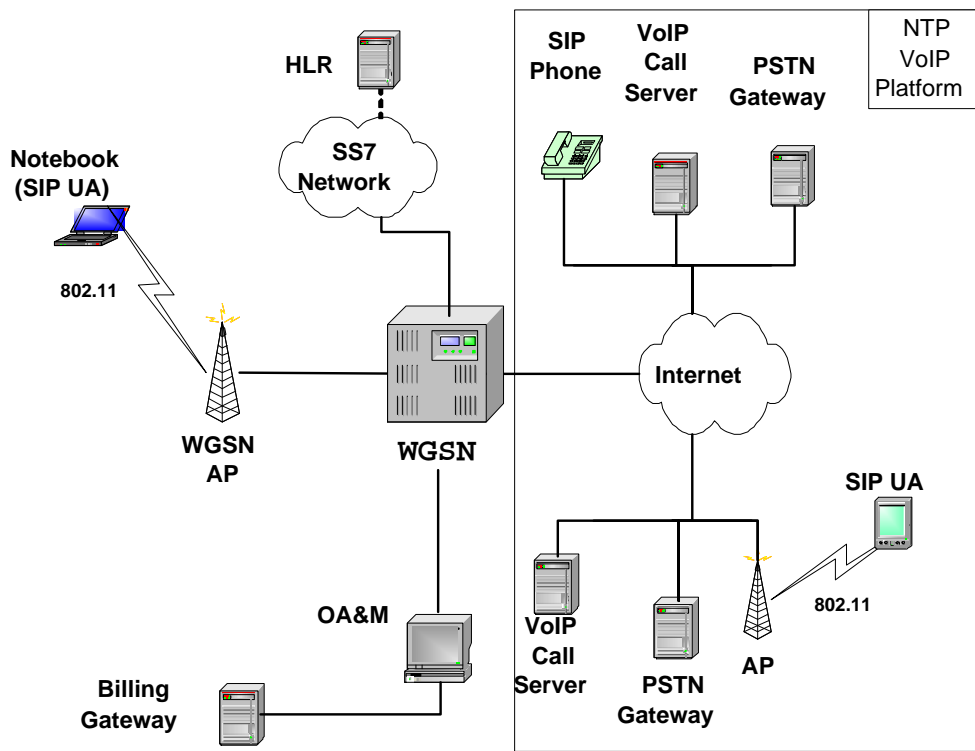
Figure 12: WGSN Architecture

Figure 12 illustrates the inter-connection between the NTP VoIP testbed and the WGSN. The WLAN radio network includes 802.11-based *Access Points* (APs) that provide radio access for the MSs. The WGSN acts as a gateway between the NTP VoIP testbed and the WLAN APs, which obtains the IP address for a *Mobile Station* (MS) from a *Dynamic Host Configuration Protocol* (DHCP) server. WGSN routes the packets between the MS and the external data networks (i.e., the NTP VoIP testbed). The WGSN node communicates with the HLR to support GPRS mobility management following 3GPP Technical Specification 23.060 [2]. Therefore, the WLAN authentication and network access procedures are exactly the same as that for GPRS described in Section 3.1.

Based on the seven interworking aspects listed in 3GPP Technical Specification 22.934 [3], we describe the following features implemented in WGSN [11].

Service aspects: WGSN provides general Internet access and VoIP services based on SIP. A *Network Address Translator* (NAT) is built in the WGSN node for IP address translation of IP and UDP/TCP headers. However, the NAT does not translate the IP address in SIP/SDP messages, and the VoIP voice packets delivered by the RTP connection cannot pass through

the WGSN node. This problem is resolved by implementing a SIP ALG in the WGSN node, which interprets SIP messages and modifies the IP address contained in these SIP messages.

Access control aspects: WGSN utilizes the standard GPRS access control for users to access WLAN services. Our mechanism reuses the existing *Subscriber Identity Module* (SIM) card and the subscriber data records in the HLR. Therefore, the WGSN customers do not need a separate WLAN access procedure, and the maintenance for customer information is simplified. User profiles for both GPRS and WLAN are combined in the same database (i.e., the HLR).

Security aspects: WGSN utilizes the existing GPRS authentication mechanism [17]. That is, the WLAN authentication is performed through the interaction between an MS (using SIM card) and the Authentication Center. Therefore, WGSN is as secured as existing GPRS networks. We do not attempt to address the WLAN encryption issue [16]. It is well known that WLAN based on IEEE 802.11b is not secured. For a determined attack, *Wired Equivalent Privacy* (WEP) is not safe, which only makes a WLAN network more difficult for an attacker to intrude. The IEEE 802.11 Task Group I is investigating the current 802.11 MAC security. For authentication, 802.1x and its *Extensible Authentication Protocol* (EAP) is the recommended approach. WGSN will follow the resulting solution.

Roaming aspects: WGSN provides roaming between GPRS and WLAN. We utilize the standard GPRS mobility management mechanism without introducing any new roaming procedures.

Terminal aspects: A WGSN MS is installed with a *Universal IC Card* (UICC) reader (a smart card reader implemented as a standard device on the Microsoft Windows platform). The UICC reader interacts with the SIM card to obtain authentication information for WGSN attach procedure. The SIP UA is implemented at the application layer of a WGSN MS.

Naming and addressing aspects: The WGSN user identification follows the *Network Access Identification* (NAI) format [4] specified in the 3GPP recommendation [3]. Specifically, the *International Mobile Subscriber Identity* (IMSI) is used as WGSN user identification.

Charging and billing aspects: The WGSN acts as a router, which can monitor and control all traffics for the MSs. The WGSN node provides both offline charging and online charging (for pre-paid services) based on the *Call Detail Records* (CDRs) delivered to the charging gateway.

20

In addition to the seven aspects listed above, WGSN also provides automatic WLAN network configuration recovery. A WGSN MS can be a notebook, which is used at home or office with different network configurations. The network configuration information includes IP address, subnet mask, default gateway, WLAN *Service Set Identifier* (SSID), etc. When the MS enters the WGSN service area, its network configuration is automatically reset to the WGSN WLAN configuration if the MS is successfully authenticated. The original network configuration is automatically recovered when the MS detaches from the WGSN. This WGSN functionality is especially useful for those users who are unfamiliar with network configuration setup. The SIP registration and call setup procedures for WGSN are given in Section 2.3. Details of WGSN implementation can be found in [11].

4   Conclusions

Under the *National Telecommunication Development Program* (NTP), we have deployed a SIP-based wireless VoIP testbed. Users participated in this testbed can freely download the SIP-based softphone client (UA) and install it on their personal computers with Windows 2000/XP operating systems. The UA provides digit-number dialing, SIP URI dialing and E.164 dialing. It also supports functions such as call hold/retrieval and call transfer. At the time of writing, the PDA version of UA program has not been released due to intellectual property issue.

As described in Section 3, NTP VoIP testbed has been integrated with GPRS and WLAN to support wireless VoIP services. In the current stage, we are integrating UMTS IP Multimedia Subsystem [17] with the NTP VoIP testbed, and will develop a wireless VoIP service network based on *Open Service Access* (OSA) specification [1]. Three extra sites are being included in the NTP VoIP testbed. They are located in National Cheng Kung University (NCKU) in Tainan, Providence University (PU) in Taichung, and Dong Hwa University (HDHU) in Hualien. The connectivity of the seven NTP VoIP sites is illustrated in Figure 13.

Acknowledgement
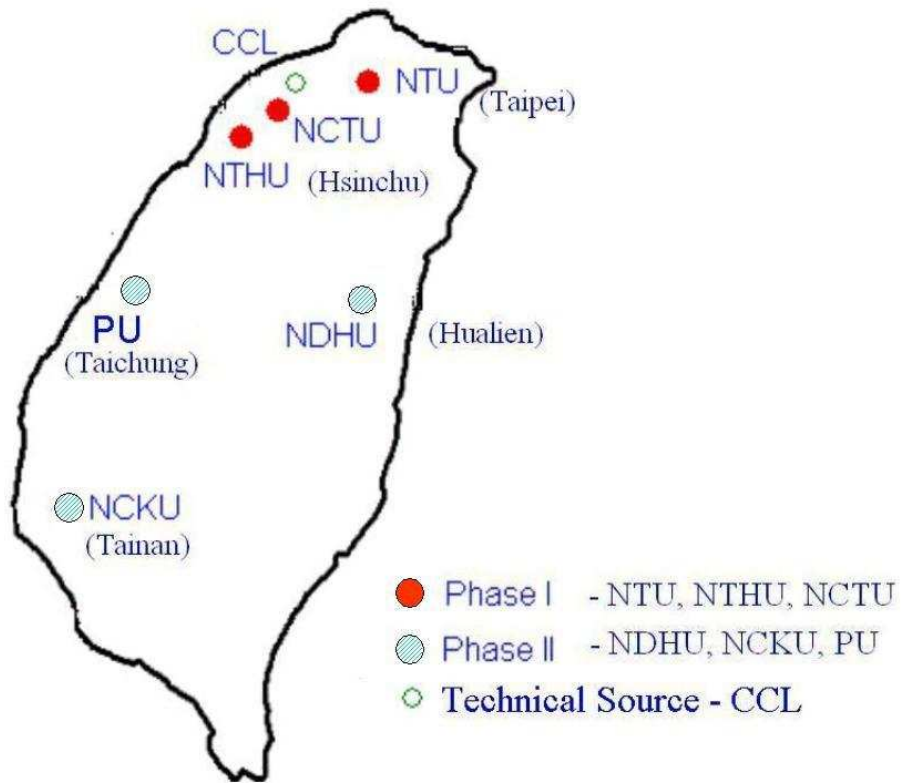
Figure 13: Connectivity of NTP VoIP Sites

References

[1] 3GPP. 3rd Generation Partnership Project; Technical Specification Core Network; Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview; (Release 4). Technical Specification 3G TS 29.198 V4.1.0 (2001-06), 2001.

[2] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service Descripton; Stage 2. Technical Specification 3G TS 23.060 version 4.1.0 (2001-06), 2001.

[3] 3GPP. 3rd Generation Partnership Project; Feasibility Study on 3GPP System to Wireless Local Area Network (WLAN) Interworking. Technical Specification 3G TS 22.934 version 6.0.0, 2002.

[4] Aboba, B., and Beadles, M. The Network Access Identifier. IETF RFC 2486, January 1999.

[5] CCITT. Numbering Plan for the ISDN Era. Technical Report Recommendation E.164 (COM II-45-E), ITU-T, 1991.

[6] Chou, S.L. and Lin, Y.-B. Computer Telephony Integration and Its Applications. *IEEE Communications Surveys*, 4(1), 2000.

[7] Daniel Collins. *Carrier Grade Voice Over IP*. McGraw-Hill, Singapore, 2001.

[8] ETSI/TC. European Digital Cellular Telecommunications System (Phase 2+); GPRS Base Station System (BSS) - Serving GPRS Support Node (SGSN) Interface; Gb Interface Layer 1. Technical Report Recommendation GSM 08.14 Version 6.0.0, ETSI, 1997.

[9] ETSI/TC. GPRS Service description Stage 2. Technical Report Recommendation GSM GSM 03.60 Version 7.0.0 (Phase 2+), ETSI, 1998.

[10] ETSI/TC. GPRS Tunnelling Protocol (GTP) across the Gn and Gp Interface. Technical Report Recommendation GSM GSM 09.60 Version 7.0.0 (Phase 2+), ETSI, 1998.

[11] Feng, V. W.-S., Wu., L.-Y., Lin, Y.-B., and Chen, W.E. WGSN: WLAN-based GPRS Environment Support Node with Push Mechanism. Accepted and to appear in *The Computer Journal*, 2004.

[12] Handley, M., and Jacobson, V. SDP: Session Description Protocol. IETF RFC 2327, April 1998.

[13] ITU. Packet-based Multimedia Communications Systems. Technical Report ITU-T H.323, Version 3, International Telecommunication Union, 1999.

[14] ITU-T. Pulse code modulation (PCM) of voice frequencies, 1988. Recommendation G.711.

[15] Li, B., et. al. QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges. *IEEE Communications Magazine*, 38(4), 2000.

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

[17] Lin, Y.-B., Hanug, Y.-R., Pang, A.-C., and Chlamtac, I. All-IP Approach for UMTS Third Generation Mobile Networks. *IEEE Network*, 16(5):8–19, 2002.

[18] Lin, Y.-B., Huang, Y.-R., Chen, Y.-K., and Chlamtac, I. Mobility Management: From GPRS to UMTS. *Wireless Communiations and Mobile Computing*, 1(4):339–360, 2001.

[19] Pang, A.-C., Lin, P., and Lin, Y.-B. Modeling Mis-routing Calls Due to User Mobility in Wireless VoIP. *IEEE Communications Letters*, 4(12):394–397, 2001.

[20] Rao, H. C.-H., Lin, Y.-B., and Chou, S.-L. iGSM: VoIP Service for Mobile Networks. *IEEE Communications Magazine*, 4(38):62–69, 2000.

[21] Rosenberg, J. et al. SIP: Session Initiation Protocol. IETF RFC 3261, June 2002.

[22] Schulzrinne, H., et al. RTP: A Transport Protocol for Real-Time Applications. IETF RFC 1889, January 1996.

[23] Sinnreich, H., and Johnston, A.B. *Internet Communications Using SIP*. John Wiley & Sons, New York, 2001.