

Design of SIP Application Level Gateway for IPv6 Translation

Whai-En Chen, Quincy Wu, Yi-Bing Lin, Yung-Chieh Lo
 Department of Computer & Information Engineering
 National Chiao Tung University
 Hsinchu, Taiwan
 {wechen, solomon, liny, yjlou}@csie.nctu.edu.tw

Abstract

This article proposes an *Application Level Gateway* (ALG) to support IPv6-IPv4 interworking for SIP applications. We describe four scenarios to deploy the SIP-based VoIPv6 applications. Based on these scenarios, we provide detailed message flows to show how SIP messages can be translated by the SIP-ALG and transparently routed between IPv6 and IPv4 networks.

Keywords: ALG, IPv6, NAT-PT, SIP, VoIP

1 Introduction

Internet research and development are a major trend in telecommunications today. Many applications have been developed for IP (*Internet Protocol*) networks. Among them, *Voice over IP* (VoIP) is one of the most important applications. Several standards have been developed for delivering voice packets over the IP networks, including H.323 specified by *International Telecommunication Union - Telecommunication* (ITU-T) [3], *Session Initiation Protocol* (SIP) [8] and *Media Gateway Control Protocol* (MGCP) [4] standardized by *Internet Engineering Task Force* (IETF). Recently, SIP has become a major approach for VoIP signaling. SIP is a text-encoded protocol based on the client-server model, which is easy to implement, extend, and can be conveniently integrated with Internet applications.

In an ideal VoIP network, every VoIP user is assigned a public IP address. However, current 32-bit IPv4 address space is limited, and it is not practical to support increasing VoIP population with public IPv4 addresses. Therefore, private IPv4 addresses are utilized in the “local environments” (i.e., private networks), and the *Network Address Translation* (NAT) technique has been developed to map several private IP addresses into one public IP address so that multiple outgoing connections can be supported by using this public IP address. A private network may not support incoming connections because the NAT does not provide private/public address mapping when the hosts in the public network initiate connections to the hosts in the private network. In this case, the public network

cannot directly routes packets to the private network. With the De-Militarized Zone mechanism, a NAT can maintain a small number of public IPv4 addresses that are statically mapped to the same number of private IPv4 addresses. Only the hosts (such as WWW, FTP or TELNET servers) assigned these private IPv4 addresses can receive incoming requests from the public IP network. On the other hand, the above issue can be gracefully resolved by using the *next generation Internet Protocol* (IP version 6, IPv6/IPng) that provides 128-bit address space. In this solution, every VoIP host is assigned a public IPv6 address, and peer-to-peer VoIP services can be offered without involving NAT. During deployment of SIP-based VoIPv6 applications, it is important to ensure that these applications can interwork with their counterparts in IPv4 networks. For example, an IPv6 user should be able to make VoIP phone calls to an IPv4 user. To support IPv6-IPv4 interworking, protocol translation is required. IETF proposes *Stateless IP/ICMP Translation* (SIIT) [7] and *Network Address Translation and Protocol Translation* (NAT-PT) [6] to support IPv6 users for accessing the IPv4-based Internet contents (such as WWW, FTP, and Email). SIIT specifies the IP and *Internet Control Message Protocol* (ICMP) header translation algorithm. The NAT-PT performs address translation for transparent routing between IPv6 and IPv4 nodes. It also serves as the default gateway of an IPv6 network. If the NAT-PT owns N public IPv4 addresses, then with additional port translation, it can simultaneously offer $N \times 65535$ IPv6 hosts to access IPv4 networks.

SIP-based VoIP calls are set up through *Session Initiation Protocol/Session Description Protocol* (SIP/SDP), and the voice data are delivered by using *Real-time Transport Protocol/Real-time Transport Control Protocol* (RTP/RTCP). Some fields in the SIP/SDP message contain IP address and transport port information. Although the NAT-PT performs IP header and TCP/UDP port translation¹, it leaves the SIP/SDP message unchanged. Therefore, a *SIP Application Level Gateway* (ALG) is required to handle the IPv6-IPv4 translation for SIP/SDP (referred to as the “*SIPv6 translation*” in this article). Several network

¹ In IETF RFC2766 [6], NAT-PT with port translation is referred to as NAPT-PT. In this article, we use the term NAT-PT.

architectures for SIPv6 translation have been proposed [1][5]. Nevertheless, the SIPv6 translation mechanisms and the relationship between the SIP servers² and the NAT-PT are seldom elaborated in depth in the literature. In this article, we focus on the design of SIP-ALG, and show how a SIP-ALG collaborates with the NAT-PT and SIP servers to support IPv6-IPv4 interworking.

2 Related Work

Figure 1 illustrates the network architecture for IPv6 translation with *Domain Name Service* (DNS) resolution. In this figure, the NAT-PT performs translation at the IP/transport layer while the *Application Level Gateway* (ALG) performs translation at the application level. An important ALG function is IPv6 translation for DNS messages (referred to as DNS-ALG). Instead of using the lengthy numerical IP address, a user typically utilizes the domain name to access an IP host. Before delivering a packet, the domain name must be resolved into the IP address by a DNS server. An IPv4 DNS server (DNS2) maintains type 'A' resource records, which resolves domain names into IPv4 addresses. Similarly, an IPv6 DNS server (DNS1) maintains resource records of type 'AAAA' or 'A6', which resolves domain names into IPv6 addresses [2]. When an IPv6 (IPv4) DNS query/response passes through the NAT-PT, the DNS-ALG appropriately modifies the message and then forwards it to the corresponding IPv4 (IPv6) DNS server.

In Figure 1, UA1 is an IPv6 host with the address 3ffe:3600:1::3. UA2 is an IPv4 host with the address 140.113.87.2. DNS1 maintains the type 'AAAA' resource record for UA1 (i.e., "ua1.ipv6.nctu.edu.tw IN AAAA 3ffe:3600:1::3"), and DNS2 maintains the type 'A' resource record for UA2 (i.e., "ua2.ipv4.nctu.edu.tw IN A 140.113.87.2"). To perform DNS resolution, a host first queries its local DNS server. If the local DNS server cannot find the corresponding resource record, it forwards the query to the root DNS server. In Figure 1, we assume that DNS1 always forwards the IPv4 DNS queries to DNS2, and DNS2 always forwards IPv6 DNS queries to DNS1. DNS1 is pre-assigned an IPv4 address 140.113.87.51. This IPv4 address is known to DNS2. On the other hand, the IPv4 address of DNS2 (i.e., 140.113.87.1) is mapped into the IPv6 address 3ffe:3600:2::140.113.87.1 at the NAT-PT. This mapped IPv6 address is known to DNS1. The NAT-PT maintains the IPv6-IPv4 address mappings for both DNS1 and DNS2.

In our example, the NAT-PT maintains a pool of public IPv4 addresses ranging from 140.113.87.51 to 140.113.87.60. These addresses are assigned to IPv6 hosts (e.g., DNS1 and UA1) when they communicate with the hosts in the IPv4 network. Also, the NAT-PT prefix 3ffe:3600:2::/96 is used to represent the "subnet" for all IPv4 hosts. The IPv6 address of an IPv4 host (e.g. DNS2 or UA2) is generated by concatenating the NAT-PT prefix with the IPv4 address of that IPv4 host.

Based on the network architecture in Figure 1, we describe an IPv6-initiated session from UA1 to UA2. As illustrated in Figure 2, the message flow is described in the following steps:

- Step 1.1.** When command "ping6 ua2.ipv4.nctu.edu.tw" is issued, UA1 sends the DNS query with 'AAAA' type to DNS1.
- Step 1.2.** DNS1 cannot find the corresponding resource record in its database. This query is forwarded to DNS2. In the IP header of this forwarded query, the source address is DNS1's IPv6 address 3ffe:3600:1::2 and the destination address is DNS2's IPv6 address 3ffe:3600:2::140.113.87.1. Since DNS1 and DNS2 reside in different subnets (3ffe:3600:1::/64 and 3ffe:3600:2::/96, respectively), DNS1 routes this query to its default gateway (i.e., the NAT-PT).
- Step 1.3.** The NAT-PT checks the IP/transport-layer headers of the DNS query. Since the transport port number is 53

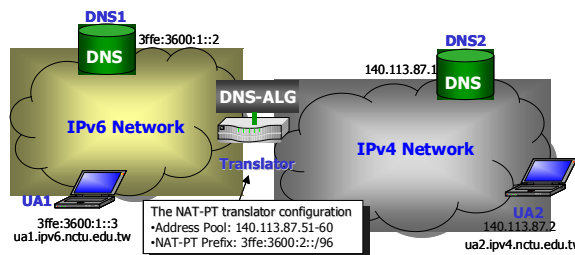


Figure 1 Network Architecture for IPv6 Translation

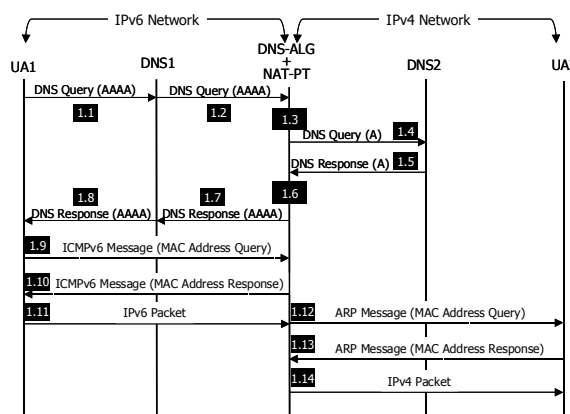


Figure 2 IPv6-initiated Session

² In this article, a SIP server represents a server with SIP registrar, proxy, and location functions [8].

(which is reserved for DNS service), the NAT-PT

passes this message to the DNS-ALG. The DNS-ALG modifies the DNS query type from ‘AAAA’ to ‘A’ and returns the modified message to the NAT-PT. The NAT-PT translates the DNS1’s IPv6 address into the IPv4 address 140.113.87.51. The DNS2’s IPv6 address 3ffe:3600:2::140.113.87.1 is translated into IPv4 address 140.113.87.1 by removing the NAT-PT prefix 3ffe:3600:2::/96.

- Step 1.4.** The translated IPv4 DNS query is sent to DNS2.
- Step 1.5.** DNS2 resolves the domain name of UA2 (i.e., ua2.ipv4.nctu.edu.tw) into the IPv4 address 140.113.87.2. This IPv4 address is included in an IPv4 DNS response. The response is returned to DNS1.
- Step 1.6.** The NAT-PT intercepts the DNS response and passes it to the DNS-ALG. The DNS-ALG translates the DNS response type from ‘A’ to ‘AAAA’. Furthermore, the DNS-ALG instructs the NAT-PT to generate the UA2’s IPv6 address by concatenating the NAT-PT prefix 3ffe:3600:2::/96 with the IPv4 address 140.113.87.2. The resulting IPv6 address is 3ffe:3600:2::140.113.87.2. The NAT-PT performs source (DNS2) and destination (DNS1) IP address translations for this message.
- Step 1.7.** The NAT-PT forwards the translated IPv6 DNS response to DNS1.
- Step 1.8.** DNS1 sends the IPv6 DNS response to UA1. At this moment, UA1 has obtained the IPv6 address of UA2.
- Steps 1.9 and 1.10 (optional).** If UA1 does not know the MAC address of the NAT-PT, it acquires this MAC address through the ICMPv6 **Router Solicitation** (with the multicast address FF02::02 defined in RFC 2373 [11]) and **Router Advertisement** message exchange [13].
- Step 1.11.** After obtaining the MAC address of the NAT-PT, UA1 sends the ICMPv6 packet (i.e., the “ping6” command) to UA2 through the NAT-PT.
- Steps 1.12 and 1.13 (optional).** If the NAT-PT does not have the UA2’s MAC address, it broadcasts the **Address Resolution Protocol (ARP)** query to request this MAC address. UA2 will reply the **ARP** response with its MAC address.
- Step 1.14.** The NAT-PT translates the source address (for UA1) from 3ffe:3600:1::3 to 140.113.87.52. The translated IPv4 address is selected from the NAT-PT’s address pool. The NAT-PT also translates the destination address (for UA2) from 3ffe:3600:2::140.113.87.2 to 140.113.87.2 by removing the NAT-PT prefix. Finally the packet is sent to UA2.

3 SIPv6 Translation

Figure 3 illustrates four scenarios to deploy the SIP-based VoIPv6 applications. In these scenarios, SIP-ALG is required to perform SIPv6 translations for SIP messages traveling between the IPv6 and IPv4 networks.

Scenario 1. The SIP *User Agents* (UAs) exchange SIP messages without involving any SIP server. The SIP-ALG translates the SIP messages delivered between the SIPv6 and SIPv4 UAs.

Scenario 2. All SIP UAs register to a SIPv4 server (the registrar function). The SIP messages exchanged between UAs are routed through this SIPv4 server. The SIP messages issued from a SIPv6 UA should be translated by the SIP-ALG before they can be delivered to the SIPv4 server. This scenario is likely to be deployed when the SIPv4 users dominate the VoIP population.

Scenario 3. The SIPv4 UAs register to an out-bound SIPv4 server and the SIPv6 UAs register to an out-bound SIPv6 server. The NAT-PT maintains IPv6-IPv4 address mappings for the SIPv6 and SIPv4 servers. When SIP messages are exchanged between the SIPv4 and SIPv6 UAs, they are first sent to their out-bound SIP servers. The SIP servers exchange the messages through the SIP-ALG for SIPv6 translation. Finally the SIP servers forward the translated messages to the destination UAs.

Scenario 4. When the SIPv6 UAs dominate the VoIP population, all SIP UAs register to a SIPv6 server. The SIP messages exchanged between UAs are routed through this SIPv6 server. For the SIP messages issued from a SIPv4 UA, the SIP-ALG performs SIPv6 translations before they are sent to the SIPv6 server.

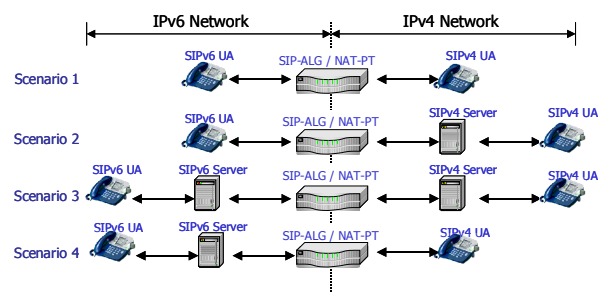


Figure 3 SIP-based VoIPv6 Scenarios

Table 1 The IP Addresses, Domain Names and Phone Numbers

Host	IPv4 address	IPv6 address	Domain Name	Phone Number
UA1	Dynamically selected at Step 2.2 (140.113.87.52)	3ffe:3600:1::3	ua1.ipv6.nctu.edu.tw	1234
UA2	140.113.87.2	3ffe:3600:2::140.113.87.2	ua2.ipv4.nctu.edu.tw	5678
UA3	Dynamically selected at Step 2.2 (140.113.87.53)	3ffe:3600:1::4	Not specified	3456
SIPv4 Server	140.113.87.40	3ffe:3600:2::140.113.87.40	sip.ipv4.nctu.edu.tw	Not specified
NAT-PT/SIP-ALG	140.113.87.50	3ffe:3600:1::1	Not specified	Not specified

Before SIPv6 translation is described, we introduce several SIP terms. A *SIP transaction* consists of a request and one or more responses. This transaction is initiated by an *initiator*. The *target* of the transaction may or may not be the *recipient* of the request. For the registration transaction example in Section 3.1, a UA (the initiator) registers to a SIP registrar (the recipient) for another UA (the target). For the invite transaction in Section 3.2, a UA (the initiator) sets up a call to another UA (the target as well as the recipient).

Consider a SIP message sent from UA1 to UA2, where UA1 is the initiator of the transaction and UA2 is the target and recipient. The addresses of UA1 and UA2 are listed in Table 1. In this message, the SIP-ALG may modify the following IP-related SIP/SDP fields:

- The *Request-URI* field (e.g., sip:5678@sip.ipv4.nctu.edu.tw or sip:5678@140.113.87.40) indicates the Uniform Resource Identifier (URI) [RFC 2396] of the receiver (UA2).
- The *To* field (e.g., To: UA2 <sip:5678@sip.ipv4.nctu.edu.tw> or To: UA2 <sip:5678@140.113.87.40>) contains an optional display name of the transaction target (which could be a person or a system, e.g., UA2) followed by its SIP URI (e.g., sip:5678@sip.ipv4.nctu.edu.tw).
- The *From* field (e.g., From: UA1 <sip:1234@sip.ipv4.nctu.edu.tw>) contains an optional display name of the transaction initiator (UA1) followed by its SIP URI sip:1234@sip.ipv4.nctu.edu.tw.
- The *Via* field (e.g., Via: SIP/2.0/UDP 140.113.87.52:5061) contains the version (SIP/2.0) and the transport protocol (UDP) followed by the IP address/port number (140.113.87.52:5061) of the immediate sender (e.g., UA1). Any intermediate server that forwards the SIP message adds a *Via* field with its address and port number. This field may also be expressed with a domain name (e.g., Via: SIP/2.0/UDP ua1.ipv6.nctu.edu.tw:5061).
- The *Contact* field (e.g., Contact: <sip:1234@ua1.ipv6.nctu.edu.tw:5061> or Contact: <sip:1234@140.113.87.

52:5061>) indicates the contact (e.g., UA1) where the other party (e.g., UA2) can send subsequent requests.

- The *Content-Length* field counts the SIP message body in octets. A *Content-Length* 0 indicates that there is no message body.

The SIP-ALG also translates the SDP fields [9][10]. These fields include:

- The *o* field (e.g., o=1234 625106937 625106937 IN IP4 140.113.87.52) indicates the originator, which contains user name (i.e., the phone number 1234), session id and version (625106937), network type (IN for Internet), address type (IP4 for IPv4, IP6 for IPv6, and FQDN for domain name), and address (140.113.87.52). The originator of the SDP is the sender of the SIP message.
- The *c* field (e.g., c= IN IP4 140.113.87.52) indicates the connection information for media (voice or data) session, which includes the network type (IN), address type (IP4), and connection address (which can be the originator's IP address 140.113.87.52). This field may also be expressed with a domain name (e.g., c= IN FQDN ua1.ipv6.nctu.edu.tw).
- The *m* field (e.g., m= audio 9000 RTP/AVP 0 4 8) indicates the media, which contains the media type (audio), port (9000), protocol (RTP/AVP), and the codec number (e.g., 0 for u-law PCM, 4 for GSM and 8 for a-law PCM [14]).

The SIP-ALG translates the *Request-URI*, *Contact*, *To*, *Via*, and *c* fields if they contain IP addresses. The port number in the *m* field may be modified (see **Step 3.5** in Section 3.2). The *Content-Length* field is re-calculated for modifications made to the SIP message body (e.g., SDP). The *From* and *o* fields are not changed.

In the remainder of this section, we describe the message flows of SIP transactions for scenario 2.

3.1. SIP Registration

In scenario 2, the registrar records for all SIP UAs are maintained in the SIPv4 server. A SIP UA can register for

another UA. In most cases, a UA registers for itself. Figure 4 illustrates the message flow where a SIPv6 UA UA3 registers for a SIPv6 UA UA1. The IPv4/IPv6 addresses and the phone numbers of the involved hosts are listed in Table 1.

The SIP registration transaction consists of a message pair exchange where the initiator is UA3, the target is UA1, and the recipient of the registration request is the SIPv4 server. In these messages, the *To* field is sip:1234@sip.ipv4.nctu.edu.tw (for UA1). The *From* field is sip:3456@sip.ipv4.nctu.edu.tw (for UA3). There is no message body (such as SDP) in these messages, and the *Content-length* field is set to 0. The following steps are executed:

Step 2.1. UA3 sends the **REGISTER** message to the SIPv4 server. The *Request-URI* is sip:sip.ipv4.nctu.edu.tw (i.e., the receiver of this message is the SIPv4 server). UA3 obtains the IPv6 address of the SIPv4 server (3ffe:3600:2::140.113.87.40) through DNS resolution (see **Steps 1.1-1.8** in Section 2). The *Contact* field is sip:1234@[3ffe:3600:1::3]:5060 (for UA1), which indicates the current location of UA1 (to be used in **Step 2.3**). The *Via* field is SIP/2.0/UDP [3ffe:3600:1::4]:5060 (for UA3). Following **Steps 1.9-1.11** in Section 2, this message is routed to the NAT-PT.

Step 2.2. The **REGISTER** message arrives at the NAT-PT. Since the destination port number is 5060 (for SIP), this message is passed to the SIP-ALG. The SIP-ALG translates UA1's IPv6 address 3ffe:3600:1::3 in the *Contact* field into the IPv4 address. In this translation, the NAT-PT is instructed to select one public IPv4 address 140.113.87.52 from its IPv4 address pool (see **Step 1.14** in Section 2). To support multiple SIPv6 UAs using one public IPv4 address, the SIP-ALG maps the default port 5060 to a free port 5061. The NAT-PT maintains the IPv6-IPv4 address mapping of UA1 (which will be used at **Step 3.3** in Section 3.2), and the *Contact* field is replaced by sip:1234@140.113.87.52:5061. The IPv6 address/port [3ffe:3600:1::4]:5060 in the *Via* field is translated into

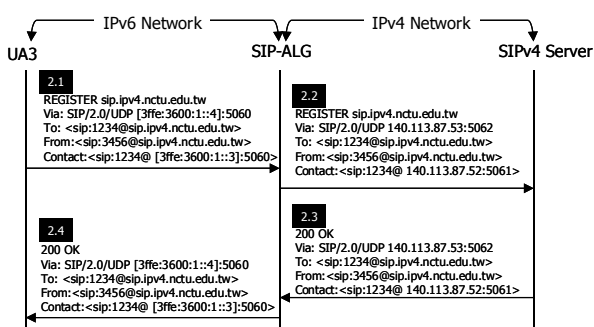


Figure 4 SIPv6 UA Registration

the IPv4 address/port 140.113.87.53:5062 (note that we assign a free port 5062 to UA3). This IPv4 address/port (for UA3) is selected from the NAT-PT's address pool. The address mapping for UA3 is maintained by the NAT-PT and will be used at **Step 2.4**. After IP-layer translation, the source address is 140.113.87.53, and the destination address is 140.113.87.40. Following **Step 1.12-1.14**, the **REGISTER** message is sent to the SIPv4 server.

Step 2.3. Upon receipt of the **REGISTER** message, the SIPv4 server retrieves UA1's registrar record based on the *To* field (i.e., the phone number 1234). From the *Contact* field, UA1's contact information (i.e., sip:1234@140.113.87.52:5061) is stored in its registrar record. Then the SIPv4 server generates the **200 OK** message by directly copying the *To*, *From*, and *Via* fields from the **REGISTER** message. The *Contact* field is sip:1234@140.113.87.52:5061, which is obtained from UA1's registrar record. In the IP header, the source address is 140.113.87.40 (for the SIPv4 server). The destination address is 140.113.87.53 (for UA3) according to the *Via* field. This message is then routed to the NAT-PT.

Step 2.4. The NAT-PT passes the **200 OK** message to the SIP-ALG. The SIP-ALG maps the IPv4 address/port 140.113.87.52:5061 in the *Contact* field to the IPv6 address/port [3ffe:3600:1::3]:5060. Note that this IPv6-IPv4 address mapping was constructed in **Step 2.2**. The NAT-PT translates the source address (for SIPv4 server) into the IPv6 format. The destination IP address/port (for UA3) is translated from 140.113.87.53:5062 to [3ffe:3600:1::4]:5060. Then the **200 OK** message is sent to UA3, and the registration procedure is complete.

In the most general case, UA1 registers for itself, where the registration message flow is the same as that in Figure 4 except that UA3 related addresses/URI in the SIP messages are replaced by those for UA1.

3.2. IPv4-to-IPv6 SIP Session Setup

After registration, a SIPv4 UA UA2 initiates an invite transaction to UA1 by sending an **INVITE** message to the SIPv4 server. In this transaction, the initiator is UA2 and the target (also the recipient of the invite request) is UA1. The message flow is illustrated in Figure 5, and details of the **INVITE**, **200 OK** and **ACK** messages are elaborated. Other provisional responses such as **100 Trying** and **180 Ringing** delivered in the invite transaction are only briefly mentioned here. The SIPv6 translations for these provisional responses are the same as the **200 OK** message except that the SIP message body is empty. For all messages exchanged in our example, the *To* field is

sip:1234@sip.ipv4.nctu.edu.tw (for UA1), and the *From* field is sip:5678@sip.ipv4.nctu.edu.tw (for UA2). The following steps are executed:

Step 3.1. UA2 sends the **INVITE** message to the SIPv4 server. In this message, the *Request-URI* is sip:1234@sip.ipv4.nctu.edu.tw (for UA1) and the *Via* field is SIP/2.0/UDP 140.113.87.2:5060 (for UA2). The *Contact* field is sip:5678@sip.ipv4.nctu.edu.tw (for UA2). In the SDP portion, the *o* field is 5678 625106937 625106937 IN IP4 140.113.87.2, the *c* field is IN IP4 140.113.87.2, and the *m* field is audio 9000 RTP/AVP 0 4 8. UA2 retrieves the destination of the next hop (i.e., sip.ipv4.nctu.edu.tw for the SIPv4 server) from the *Request-URI*. The retrieved domain name is translated into the IPv4 address 140.113.87.40 through DNS resolution. Then this message is sent to the SIPv4 server.

Step 3.2. Upon receipt of the **INVITE** message, the SIPv4 server utilizes the phone number 1234 in the *Request-URI* to retrieve UA1's IP address/port number 140.113.87.52:5061. The *Request-URI* is then modified as sip:1234@140.113.87.52:5061. The SIPv4 server adds its address as the second *Via* field SIP/2.0/UDP 140.113.87.40:5060. The *Contact* field and the SDP portion are not modified. In the IP header, the source address is 140.113.87.40 (for the SIPv4 server), and the destination address (for UA1) is 140.113.87.52 (from the *Request-URI*). The message is routed to the NAT-PT.

Step 3.3. The NAT-PT passes the **INVITE** message to the SIP-ALG. The SIP-ALG translates the *Request-URI* into the IPv6 format (sip:1234@[3ffe:3600:1::3]:5060) based on the mapping constructed in **Step 2.2**. The SIP-ALG translates the second *Via* field into SIP/2.0/UDP [3ffe:3600:2::140.113.87.40]:5060. In the SDP portion, the SIP-ALG translates the *c* field IN IP4 140.113.87.2 into the IPv6 format (IN IP6

changed. The NAT-PT translates the IPv4 source address (for SIPv4 server) into the IPv6 format 3ffe:3600:2::140.113.87.40. The destination address (for UA1) is translated into 3ffe:3600:1::3. Then the message is sent to UA1.

Step 3.4. Assume that the call is accepted by UA1 after the **100 Trying** and **180 Ringing** have been sent out. UA1 sends the **200 OK** message to UA2. The *To*, *From*, *Via* fields are directly copied from the **INVITE** message. The *Contact* field is sip:1234@sip.ipv4.nctu.edu.tw (for UA1). In the SDP portion, the *o* field is 1234 4687790 4687790 IN IP6 3ffe:3600:1::3, the *c* field is IN IP6 3ffe:3600:1::3, and the *m* field is audio 9000 RTP/AVP 0. From the second *Via* field (note that the *Via* fields are retrieved in the reverse order as they were inserted into the **INVITE** message), UA1 obtains IPv6 address/port [3ffe:3600:2::140.113.87.40]:5060 of the next hop (i.e., the SIPv4 server). Base on this address, the **200 OK** message is routed to the NAT-PT.

Step 3.5. When the NAT-PT receives the **200 OK**, it passes this message to the SIP-ALG. The SIP-ALG translates the *c* field of SDP into IN IP4 140.113.87.52. For the *m* field, the port number 9000 is replaced by 9002 to prevent multiple SIPv6 UAs from using the same public IPv4 address and port number. The SIP-ALG builds the RTP/RTCP port mappings (9002 for RTP and 9003 for RTCP) for the subsequent voice/data sessions. The *o* field is not changed. The SIP-ALG translates the second *Via* field into the IPv4 format 140.113.87.40:5060, which is the address of the next hop (the SIPv4 server). The NAT-PT translates the IPv6 source (UA1) and destination (the SIPv4 server) addresses into IPv4 addresses 140.113.87.52 and 140.113.87.40. Then the message is sent to the SIPv4 server.

Step 3.6. Upon receipt of the **200 OK** message, the SIPv4 server removes the second *Via* field, and retrieves the IPv4 address/port 140.113.87.2:5060 (UA2) from the first *Via* field. The message is forwarded to UA2.

Steps 3.7-3.9. When UA2 receives the **200 OK** message, it starts the RTP/RTCP sessions. In the mean time, UA2 replies the **ACK** message to inform UA1 that the invite transaction is complete. **Steps 3.7-3.9** are similar to **Steps 3.1-3.3** except that the *Content-length* is set to 0.

4 Discussion and Conclusions

To support IPv6-IPv4 interworking for SIP applications, this article proposed an *Application Level Gateway* (ALG) to handle the SIP address translation (referred to as the "*SIPv6 translation*"). We described four scenarios to deploy the SIP-based VoIPv6 applications.

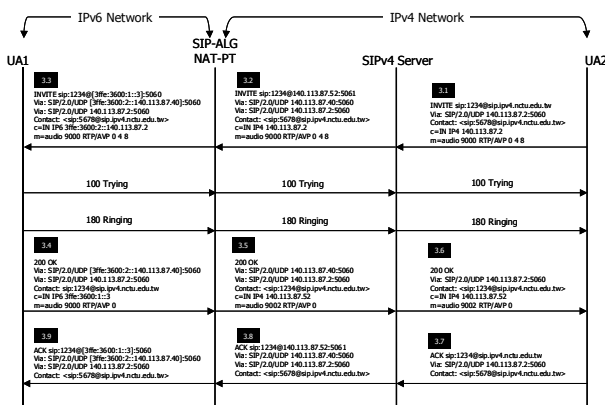


Figure 5 IPv4-to-IPv6 SIP Session Setup with the SIPv4 Server

3ffe:3600:2::140.113.87.2). The *o* and *m* fields are not

Then we provided detailed message flows and SIP-ALG translations for scenario 2.

The behaviors of the SIP-ALG for the four scenarios are similar with some exceptions. For scenarios 1 and 3, the SIP-ALG is not involved in the registration transaction. For scenario 4, the SIPv6 translation is similar to that for scenario 2, except that the IPv6-IPv4 translation directions are reversed. In scenario 1, the SIPv6 UAs can only be identified by port 5060, and therefore the number of simultaneous calls from SIPv4 UAs to SIPv6 UAs is limited by the size of the IPv4 address pool in the NAT-PT.

In summary, the SIP-ALG resolves the SIPv6 translation issues by performing IPv6-IPv4 address translation at the application level. An important future direction is to design an efficient engine to perform SIP-ALG functions.

- [1] G. Bajko, B. Bertenyi, K. Kiss. *Multimedia sessions between 3G wireless and Internet users*. IEEE ICC. 2(2): 2001, pp.435-439.
- [2] S. Thomson, C. Huitema. *DNS Extensions to support IP version 6*. IETF RFC 1886. December 1995.
- [3] ITU-T. Rec. H.323. *Packet-based multimedia communication systems*. February 1998.
- [4] F. Andreassen, B. Foster. *Media Gateway Control Protocol (MGCP) Version 1.0*. IETF RFC3435, January 2003.
- [5] P. Flykt, T. Alakoski. *SIP services and interworking with IPv6*. Proceedings of the IEE Second International Conference on 3G Mobile Communication Technologies, 2001, pp.186–190.
- [6] G. Tsirtsis, P. Srisuresh. *Network Address Translation - Protocol Translation (NAT-PT)*. IETF RFC 2766, February 2000.
- [7] E. Nordmark. *Stateless IP/ICMP Translation Algorithm (SIIT)*. IETF RFC 2765, February 2000.
- [8] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. *SIP: Session Initiation Protocol*. IETF RFC3261, June 2002.
- [9] M. Handley, V. Jacobson. *SDP: Session Description Protocol*. IETF RFC 2327. April 1998.
- [10] S. Olson, G. Camarillo, A. B. Roach. *Support for IPv6 in Session Description Protocol (SDP)*. IETF RFC 3266. June 2002.
- [11] R. Hinden, S. Deering. *Internet Protocol Version 6 (IPv6) Addressing Architecture*. IETF RFC 3513. April 2003.
- [12] R. Hinden, B. Carpenter, L. Masinter. *Format for Literal IPv6 Addresses in URL's*. IETF RFC 2732. December 1999.
- [13] T. Narten, E. Nordmark, W. Simpson. *Neighbor Discovery for IP Version 6 (IPv6)*. IETF RFC 2461. December 1998.
- [14] H. Schulzrinne. *RTP Profile for Audio and Video Conferences with Minimal Control*. Audio-Video Transport Working Group, IETF RFC 1890. January 1996.
- [15] T. Berners-Lee, R. Fielding, L. Masinter. *Uniform Resource Identifiers (URI): Generic Syntax*. IETF RFC 2396. August 1998.

References

Biographies



Whai-En Chen received a Bachelor of Science degree in Electric Engineering from TamKang University in 1997, and earned a Ph.D. in Computer Science from National Tsing Hua University in 2002. He began serving as a research assistant professor in National Chiao Tung University, and helped National Telecommunications Project Office to deploy a SIP-based VoIP Platform across several universities in 2002. His research interests include 3G IP Multimedia Subsystem (IMS), Session Initiation Protocol (SIP), Internet Protocol Version 6 (IPv6), Design and implementation of high-speed lookup and classification engines, and Service Creation on the Third Generation Mobile Network.

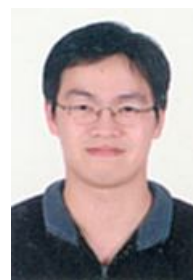


Quincy Wu received a Bachelor of Science degree in Mathematics from National Tsing Hua University in 1992, and earned a Ph.D. in Computer Science and Information Engineering from National Tsing Hua University in 2000. He joined National Center for High-Performance Computing with the NBEN (National Broadband Experimental Network) project, where he successfully designed and established the first island-wide IPv6 network among universities. In 2002, he began serving as a research assistant professor with National Chiao Tung University, and helped National Telecommunications Project Office to deploy a SIP-based VoIP Platform across several universities. His research interests focus on Session Initiation Protocol, Open Service Architecture, Internet Protocol Version 6, Design and Analysis of Approximation Algorithms, and Service Creation on the Third Generation Mobile Network.



Yi-Bing Lin received his BSEE degree from National Cheng Kung University in 1983, and his Ph.D. degree in Computer Science from the University of Washington in 1990. From 1990 to 1995, he was with the Applied Research Area at Bell Communications Research (Bellcore), Morristown, NJ. In 1995, he was appointed as a professor of Department of Computer Science and Information Engineering (CSIE), National Chiao Tung University (NCTU). In 1996, he was appointed as Deputy Director of Microelectronics and

Information Systems Research Center, NCTU. During 1997-1999, he was elected as Chairman of CSIE, NCTU. His current research interests include design and analysis of personal communications services network, mobile computing, distributed simulation, and performance modeling. Dr. Lin is an associate editor of IEEE Network, an editor of IEEE Trans. on Wireless Communications, an associate editor of IEEE Trans. on Vehicular Technology, an associate editor of IEEE Communications Survey and Tutorials, an editor of IEEE Personal Communications Magazine, an editor of Computer Networks, an area editor of ACM Mobile Computing and Communication Review, a columnist of ACM Simulation Digest, an editor of International Journal of Communications Systems, an editor of ACM/Baltzer Wireless Networks, an editor of Computer Simulation Modeling and Analysis, an editor of Journal of Information Science and Engineering, Program Chair for the 8th Workshop on Distributed and Parallel Simulation, General Chair for the 9th Workshop on Distributed and Parallel Simulation. Program Chair for the 2nd International Mobile Computing Conference, Guest Editor for the ACM/Baltzer MONET special issue on Personal Communications, a Guest Editor for IEEE Transactions on Computers special issue on Mobile Computing, a Guest Editor for IEEE Transactions on Computers special issue on Wireless Internet, and a Guest Editor for IEEE Communications Magazine special issue on Active, Programmable, and Mobile Code Networking. Lin is the author of the book Wireless and Mobile Network Architecture (co-author with Imrich Chlamtac; published by John Wiley & Sons). Lin is an Adjunct Research Fellow of Academia Sinica, Chair Professor of Providence University. He is an IEEE Fellow and an ACM Fellow.



Yung-Chieh Lo received a Master degree in Computer Science and Information Engineering from National Chiao Tung University in 2003. Currently, he is an engineer in Realtek. His research interests includes Session Initiation Protocol (SIP), Internet Protocol Version 6 (IPv6).