IPv4/IPv6 Transition for SIP VoIP Applications

Dr. Whai-En Chen
Research Assistant Professor
Department of Computer Science
National Chiao Tung University, Taiwan 300, R.O.C.
TEL: +886-3-5731924 FAX: +886-3-5716929
Email: wechen@cs.nctu.edu.tw
Outline

• Introduction
• NAT Solutions and Problems
• IPv6 SIP User Agent Development
• Socket-layer Translator
• SIPv6 Translator
• SIPv6 Analyzer
• Conclusion and Future Work
Introduction

• Since *Internet Protocol version 6* (IPv6) not only provides rich IP addresses but also support better mobility, security and *Quality of Service* (QoS) than IPv4, IPv6 has become one of the most important protocol in the next generation Internet.

• To deploy *Voice over IP* (VoIP) in Taiwan, we have *IP address shortage problem* since each VoIP device requires a public IP address. To solve this problem, IPv6 is the best choice and a long-term solution.

• On the other hand, in the early stage of IPv6 deployment, VoIP is an important driving force for IPv6 promotion.

• This presentation includes four IPv6 *Session Initiation Protocol* (SIP) VoIP applications and their deployment in Taiwan.
NAT Solutions and Problems
NAT Solution & Problem

RFC 3261: received parameter
RFC 3581: rport parameter
Static Routing
Universal Plug and Play (UPnP)

Discover the IP address of NAT

Obtain the external IP address

Create the external port number
Simple Traversal of UDP Through NATs (STUN)

Like UPnP, STUN supports automatic configuration of private-to-public IP information mappings in both the UA and the NAT before a SIP call is set up.

Three types of NAT uses source IP/port to map an external port.

Therefore, a SIP User Agent (UA) can ask the STUN server the mapped IP/port information.

STUN does not need any modification to the NAT.

STUN cannot be applied in a symmetric NAT.
Time Complexities for the NAT Traversing Mechanisms

<table>
<thead>
<tr>
<th>NAT Traversing Mechanism</th>
<th>IP Information Mapping Establishment</th>
<th>VPN Connection Establishment</th>
<th>Call Setup</th>
<th>RTP Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPN</td>
<td>N/A</td>
<td>2114 ms</td>
<td>161 ms</td>
<td>8.3 ms</td>
</tr>
<tr>
<td>Static Route</td>
<td>Manual Setup</td>
<td>N/A</td>
<td>71 ms</td>
<td>0.7 ms</td>
</tr>
<tr>
<td>UPnP</td>
<td>261 ms</td>
<td>N/A</td>
<td>71 ms</td>
<td>0.7 ms</td>
</tr>
<tr>
<td>STUN</td>
<td>27 ms</td>
<td>N/A</td>
<td>71 ms</td>
<td>0.7 ms</td>
</tr>
</tbody>
</table>

- The IP information mapping establishment for STUN is much faster than that for UPnP.
- VPN connection establishment is much slower than that for IP information mapping establishment.
- The call setup time for the VPN mechanism is more than twice of that for the other three mechanisms.
- Also, the RTP latency for the VPN mechanism is much longer than that for the other three mechanisms. The last two results are due to the VPN tunnel overhead.
IPv6 SIP User Agent Development
Some Socket APIs, parameters and data structures of IPv6 are different from that of IPv4 and should be modified.
## Socket Parameter

<table>
<thead>
<tr>
<th>IPv4</th>
<th>IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF_INET</td>
<td>AF_INET6</td>
</tr>
<tr>
<td>PF_INET</td>
<td>PF_INET6</td>
</tr>
<tr>
<td>IN_ADDR_ANY</td>
<td>inaddr6_any</td>
</tr>
</tbody>
</table>
# Data Structure (1)

<table>
<thead>
<tr>
<th>IPv4</th>
<th>IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>sockaddr</td>
<td>sockaddr_storage</td>
</tr>
<tr>
<td>in_addr</td>
<td>in6_addr</td>
</tr>
<tr>
<td>sockaddr_in</td>
<td>sockaddr_in6</td>
</tr>
<tr>
<td>hostent</td>
<td>addrinfo</td>
</tr>
</tbody>
</table>
## Data Structure (2)

<table>
<thead>
<tr>
<th>IPv4</th>
<th>IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>sin_len</td>
<td>sin6_len</td>
</tr>
<tr>
<td>sin_family</td>
<td>sin6_family</td>
</tr>
<tr>
<td>sin_port</td>
<td>sin6_port</td>
</tr>
<tr>
<td>sin_addr</td>
<td>sin6_addr</td>
</tr>
<tr>
<td>s_addr</td>
<td>s6_addr</td>
</tr>
</tbody>
</table>
# Name Resolution Functions

<table>
<thead>
<tr>
<th></th>
<th>IPv4</th>
<th>IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name to Address</strong></td>
<td>inet_aton()</td>
<td>inet_pton()</td>
</tr>
<tr>
<td></td>
<td>inet_addr()</td>
<td></td>
</tr>
<tr>
<td></td>
<td>inet_ntoa()</td>
<td>inet_pton()</td>
</tr>
<tr>
<td><strong>Address Resolution</strong></td>
<td>gethostbyname()</td>
<td>getipnodebyname()</td>
</tr>
<tr>
<td></td>
<td>gethostbyaddr()</td>
<td>getipnodebyaddr()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>getnameinfo()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>getaddrinfo()</td>
</tr>
</tbody>
</table>
Checkv4.exe (Partial Results)

- Input source codes to checkv4.exe
- Checkv4.exe outputs the file names, line numbers and hints to modify the codes to IPv6
Get Local Address

- SIP User Agent should provide the IPv4 and IPv6 addresses of the local host.
- The IPHelper functions
  - Microsoft Windows system provides this function from Windows 98
  - This solution works on both Windows XP and 2003
  - It’s a Windows-only solution
- Function name: GetAdaptersAddresses()
Parsing IPv6 URI in SIP and SDP

- IPv4 SIP URI: sip:wechen@140.113.131.12:5060
- IPv6 SIP URI: sip:wechen@[3ffe:1345:5643::3]:5060
- IPv4 parser assumes that \texttt{semicolon} is used to separate the IP address and port number, and the SIP parser in SIP and SDP protocol stacks should be modified to process IPv6 address and port number.

- IP6 address type and IPv6 address in Session Description Protocol (SDP)
  - \texttt{c=IN IP6 FE80:60::2}
GUI Problem

- IP Address Control
  - IPv4 specified
  - Do not accept domain name and IPv6

- The variable-length Input Component
An IPv6 SIP User Agent

1. Configuration
2. Dialing
3. SIP signaling (IPv6)

Using IPv6 Addresses

IPv6 Network (Showroom) → SIPv6 UA
SIPv6 UA → IPv6 Network (IPv4) → Dual-stack Router → Internet (IPv4) → IPv6 Network (NCTU VoIP Lab)

3.1 INVITE
3.2 INVITE
3.3 INVITE
3.4 200 OK
3.5 200 OK
3.6 200 OK
3.7 ACK
3.8 ACK
3.9 ACK
4. RTP

4. Video

SIPv6 UA → IPv6 Network (NCTU VoIP Lab)
IPv6 Network (NCTU VoIP Lab) → Dual-stack Router → Internet (IPv4) → IPv6 Network (Showroom)

Using IPv6 Addresses

圖例：
- Dual-stack Router
- Tunneling
- SIP signaling (IPv6)
- SIP signaling (Tunnel)
Porting IPv4 SIP UA to IPv6

• Modification
  – Socket data structures, parameters and APIs should be modified
  – User interface should be modified
  – SIP parser should be modified

• Take SIPv6 UA porting for an example
  – Total source codes are about 100,000 lines in 150 files
  – Total 600 lines in 39 files should be modified
  – Only about 300 lines can be found by checkv4.exe (provided by Microsoft)

• The porting work is tedious and not efficient.

• Based on BIA-like architecture, we proposed a Socket-layer Translator (SLT) to translate SIP VoIP applications on Windows.

BIA: Bump-In-the-API; see IETF RFC3338
Socket-layer Translator
IPv4/IPv6 Translation Mechanisms

**BIS**
Bump-In-the-Stack

**BIA**
Bump-In-the-Application
Socket-layer Translator (SLT)

Users can access IPv6 resources through IPv4 applications and SLT.
IPv4 to IPv6 Resolving Flow

1. DNS Query ‘A’
2. DNS Query ‘AAAA’
3. Reply IPv6 address
4. Request IPv4 address
5. Reply IPv4 address
6. Reply IPv4 address
7. Send data using IPv4
8. Request IPv6 address
9. Reply IPv6 address
10. Send data using IPv6
11. Reply using IPv6
12. Request IPv4 address
13. Reply IPv4 address
14. Reply using IPv4
Using IPv4 to Browse Without Socket-layer Translator
Name Resolving: Map IPv6 address to IPv4 address

```plaintext
C:\\WINDOWS\System32\nrttest www.kame.net
he's hostname:www.kame.net
it's alias names:
addrtype is 2
addr length is 4
10.128.128.127
10.128.128.126
```
Socket-layer Translator Result
Socket-layer Translator
Performance Evaluation of NCTU SLT

- Files with sizes ranging from 10KB to 2GB are downloaded from the FTP server via the 100Mbps Ethernet.
- **Pure IPv4** takes 205 seconds
- **Pure IPv6** takes 215 seconds
- **NCTU SLT** only requires 4 more seconds over **Pure IPv6** and 15 seconds over **Pure IPv4**.
SIPv6 Translator
SIPv6 Translator

- Through manual modification and Socket-layer Translator, we have IPv6 SIP UAs (SIPv6 UAs).

- However, only using SIPv6 UAs, which can utilize rich IPv6 addresses, does not solve the IP address shortage problem in VoIP deployment, because a SIPv6 UA cannot communicate with a SIPv4 UA.

- To solve this problem, we develop a SIPv6 Translator based on the architecture proposed in IETF RFC 2766 (Network Address Translation and Protocol Translation, NAT-PT).
SIPv6 Translator Architecture

BSS: Base Station Subsystem
CSCF: Call State Control Function
GERAN: GPRS Edge Radio Access Network
GGSN: Gateway GPRS Support Node
GPRS: General Packet Radio Service
HSS: Home Subscribe Server
IPv4: Internet Protocol version 4
IPv6: Internet Protocol version 6
IMS: IP Multimedia Core Network Subsystem
IMS-ALG: IMS Application Level Gateway
MS: Mobile Station
SGSN: Serving GPRS Support Node
TrGW: Translation Gateway
UA: User Agent
UE: User Equipment
UTRAN: UMTS Terrestrial Radio Access Network
IMS-ALG Architecture
Call Setup Flow From MS to IPv4 Called Party

1. INVITE 0944021210@CSCF
c=IN IP6 3FFE:3600:1::11
m=audio 9000 RTP/AVP 0

2. INVITE 0944021210@CSCF
c=IN IP6 3FFE:3600:1::11
m=audio 9000 RTP/AVP 0

3. Create Call-ID
m=receive
c=IP4 $p=$
4. 200 OK

5. INVITE 0944021210@IPv4_proxy (v4)
c=IP4 140.113.131.1
m=audio 9002 RTP/AVP 0

6. INVITE 0944021210@Callee
c=IN IP4 140.113.131.12
m=audio 8002 RTP/AVP 0

7. 200 OK

8. 200 OK
c=IN IP4 140.113.131.12
m=audio 8002 RTP/AVP 0

9. Modify Call-ID
m=send
c=IP4 140.113.131.12
p=8002
10. 200 OK

11. Create Call-ID
m=sendreceive
c=IP6 3FFE:3600:1::11
p=9000

12. 200 OK

13. 200 OK
c=IN IP6 3FFE:3600:2::1
m=audio 8000 RTP/AVP 0

14. 200 OK
c=IN IP6 3FFE:3600:2::1
m=audio 8000 RTP/AVP 0

15. ACK 0944021210@CSCF

16. ACK 0944021210@IPv4_proxy (v6)

17. ACK 0944021210@IPv4_proxy (v4)

18. ACK 0944021210@Callee

IPv6 RTP Stream
IPv4 RTP Stream
Call Termination Flow From MS to IPv4 Called Party

- UE
  - BYE 0944021210@IMS
  - 200 OK
  - 200 OK

- CSCF
  - BYE 0944021210@IPv4_proxy

- IMS-ALG
  - BYE 0944021210@IPv4_proxy
  - 200 OK
  - Terminate CallHD
  - 200 OK

- TrGW
  - a

- SIPv4 proxy
  - BYE 0944021210@Callee
  - 200 OK

- Callee
Fault Tolerance
The IPv6 SIP-based VoIP Deployment

- 0944006XXX is assigned to IPv6 network.
- 0944004XXX is assigned to IPv4 network.
- The forwarding rules are set in the SIP proxies.
The IPv6 SIP-based VoIP Deployment

Specific Functions:
1. NIC Selection
2. Packet Dissection
3. Packet Filtering Rule
4. SIP Dialog
5. SIP Message Flow
6. RTP Replay
7. Statistics
8. Jitter Buffer
9. Video Replay
IPv4/IPv6 SIP VoIP Interworking
# The Interoperability Test Results

<table>
<thead>
<tr>
<th></th>
<th>SIP Message</th>
<th>SDP Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Request URI</td>
<td>Contact</td>
</tr>
<tr>
<td><strong>IP Soft Phone</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCL Skin UA</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Windows Messenger 4.7.2009</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>IP Hard Phone</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PingTel 2.1.10</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>snom 200</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Cisco IP Phone 7940 Series</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>PSTN Gateway</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vontel PSTN Gateway</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Cisco PSTN Gateway</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

- The SIPv6 UA developed by NCTU can communicate with all of the commercial IPv4 SIP UAs through the SIPv6 Translator.
- The IPv4 SIP UAs are deploy in the NTP VoIP platform.

http://www.voip.ntpo.org.tw
SIPv6 Analyzer
SIPv6 Analyzer

Control Panel

Packet List

Protocol Parser
(using Ethereal parser)

Hex Dump
SIP Viewer

SIP Viewer automatically collect SIP messages.

Call-ID From To

SIP Dialog Collection

SIP Flowchart

SIP Flowchart from Headers
RTP Viewer

RTP Viewer can play back Video and Voice!

RTP Session List

RTP Viewer automatically collect RTP packets according to the SDP c and m fields.

Video Playback

Video and Voice Control Panel

Yueh-Hsin Sung
Conclusions

• In this presentation, we provide IPv6 solutions for VoIP.
• First, we introduce the problems of NAT solutions.
• Then, we take SIP UA as an example to show the IPv6 porting work.
• To improve the IPv6 transition progress for applications, we introduce a Socket-layer Translator solution.
• To bridge IPv4 and IPv6 VoIP, we provide the SIPv6 Translator that can translate both SIP signaling and RTP media packets from IPv4 to IPv6 and vice versa.
• Finally, we present a SIPv6 Analyzer tool that can help the IPv6 SIP VoIP development and deployment. This tool can be downloaded from our web site [5].
• Above work is sponsored by Taiwan NICI IPv6 Project and NTP VoIP Project.
References


Q & A

Thank You!