The Design of Session Border Controller for Traversing Network Address Translation on VOCAL*

LIU Fa-gui, RAO Peng
(School of Computer Science & Engineer, South China University of Technology, Guangzhou 510640, China)

Abstract: With the accomplishment of its technical maturation, VoIP has been more rapidly developed and spread. However, some difficulties and problems also arise unavoidably and the problem, how VoIP applications traverse NAT or Firewall is one of them. VOCAL is an open-source code program that provides a SIP-based VOIP communication. Therefore, NAT traversal problem must be resolved when developing VoIP services with VOCAL. This thesis analyzes the issues brought by NAT in VoIP services and discusses the main solutions for traversing NAT. And then a solution over the VOCAL system named SBC is brought forward, which can do without changing any the present device. A SBC is located at the border of VOCAL system. SBCs manipulate SIP messages and transmit media packets so as to provide traversal of NAT/Firewalls. The design of SBC on VOCAL has been done in this thesis.

Key words: VoIP; VOCAL; SIP; SBC; NAT
CLC number: TN916.2 **Document code**: A **Article ID**: 0529-6579 (2006) 06-0026-05

VoIP (Voice over Internet Protocol) is the routing of voice conversations over the Internet or any other IP-based network. In the computer multimedia technology and Internet technology promotion, VoIP possesses very significant development potential, although it's been a short time since VoIP was introduced. With the development of the broadband network and the introduction of packet-based communications controlling protocols, such as RTP, RTCP, RSVP, IPv6 and so on, the quality of voice in VoIP can be rival to the local calls. There are two signaling protocols used in VoIP: ITU H.323 and IETF SIP (Session Initiation Protocol) [1]. Because of SIP's simplicity, agility and extensibility, the research in this area has won more attention.

With the rapid expansion of Internet, the IP addresses of IPv4 with 32 bits are facing depletion situation. On the other hand, for the private network security considerations, a lot of enterprise networks using private IP addresses access Internet through NAT (Network Address Translation) export. However, in multimedia communications the trouble of traversing NAT seriously constrains IP telephony and videoconferencing applications.

For the VOCAL (Vovida Open Communication Application Library) system does not provide solutions across NAT, it is very important to settle down this problem when developing VoIP services with VOCAL. This thesis researches the VOCAL system and then a solution over the Vocal system named SBC (Session Border Controller) is brought forward. The design of SBC is also discussed.

1 VOCAL

1.1 VOCAL architecture

Figure 1 is the architecture of VOCAL. VOCAL is a system of distributed architecture, made up of several different servers. The Marshal Server (MS) is an implementation of the SIP proxy server and acts as the initial point of contact for all SIP signals that enter the VOCAL system [2]. Other servers are talked about in reference 2 and 5.

The functions and SIP message interactions of each server in the VOCAL are based on a status machine mode. After initialization, according to the mode in status machine, triggered by SIP events, the server implements the SIP by executing different operations correspond to different statuses [5].

1.2 Development on VOCAL

As an open-source code program with consummate open-source SIP stack and clear easy-modified code, VOCAL is chosen to develop VoIP applications on. Currently, the main developments based on the VOCAL are billing system development, new feature service development [4-5], traversal of NAT and network security protection.

* 收稿日期：2006-04-18
基金项目：广东省粤港关键领域重点突破基金资助项目（2005A10207005；信产厅 2004-0005）
作者简介：刘发贵（1963年生），女，副教授，在职博士；E-mail：fgliu@scut.edu.cn

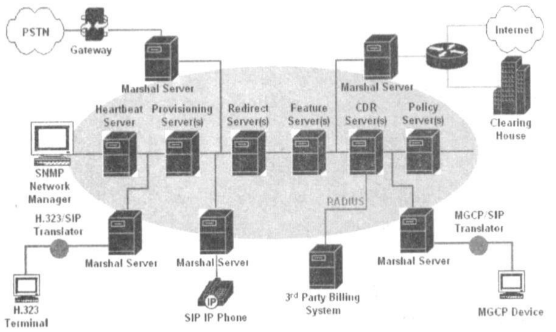

Fig. 1　VOCAL Architecture

2　NAT Issue & Solutions

2.1　NAT issue

NAT causes some problem in IP voice and video communications. NAT technology allows enterprise to use private IP addresses on the LAN. Unfortunately, routers can only send data to the routable IP addresses (public IP addresses). This leads to that terminals with public IP addresses cannot call those with private IP addresses.

Though the terminal within the NAT can call the terminal outside, there is still some problem. When establishing a call, the IP address and port, launching the calling terminal, will be included in the SIP signaling message. The called terminal will derive IP address of the calling terminal from the packet and begin sending audio and video data to the IP address of the calling terminal. If it is a private IP address which is identified to be unroutable, routers in Internet will throw away the audio and video data packets. This call will display already connected, but the calling terminal within NAT will never receive the audio and video data packets from the called terminal outside[3].

2.2　Solutions for traversing NAT

Currently, the main solutions for traversal of NAT are as follows:

(1) Application Layer Gateway (ALG)

Generally, NAT transforms IP address through amending the header of UDP and TCP packets which contains the IP address information. However, there is also IP address information in SIP messages for the VoIP application based on SIP protocol. The ALG solution is to add ALG functional modules to NAT devices, which can change the IP address information in SIP message and open an idle port according to the modified SDP (Session Description Protocol) message to achieve normal conversation between NAT and beyond.

The ALG solution causes lots of NAT/FW devices without ALG capacity to be replaced or upgraded, but users want operators to provide new service without changing the NAT devices[7-8].

(2) Simple Traversal of UDP Through Network Address Translators (STUN)

The STUN solution gets NAT's public IP address in advance, which is the address transformed by NAT according to the private IP address of VoIP User Agent (UA), through Internet communications between STUN client in private network and STUN server in public network. Then, STUN modifies the address content in signal message using the public IP address, so that NAT can transmit the UDP packets received to the right private address, namely the private address of VoIP UA.

The limitations of STUN are that VoIP UA should support STUN client, while STUN doesn't support TCP protocol. In addition STUN cannot traverse Symmetric NAT[7-8].

(3) Traversal Using Relay NAT (TURN)

The TURN solution is similar to the STUN solution. The TURN solution gets the IP address of TURN server in public address from TURN client in private network, and modifies the address content in signal message using the address TURN server transmits messages for UA, and UA exchanges messages with TURN server through Turn client.

The limitations of TURN are that VoIP UA should support TURN client, which is the same with STUN. In addition, all the messages should be transmitted by TURN server, which induces the increasing possibility of lapse and packets lost[7-8].

(4) Session Border Controller (SBC)

A SBC is a VoIP session-aware device that controls call admission at the border of the network, and optionally (depending on the device) performs as a host of call-control functions to ease the load on the call agents within the network.

A SBC can help VOCAL traverse the existing NAT/FW without any changes in VOCAL system. It is located at the border of the network device core. All the signaling messages and media packets through the SBC will be manipulated, and the modified messages can be transferred correctly between user and VOCAL system. NAT/FW accepts the modified messages and transfers them to the user.

So far, SBC is the most perfect solution for traversal of NAT. SBC needn't modify the NAT devices and softswitch in the network, and meanwhile prevent DoS (Denial of Service) attacks, enhanced QoS (Quality of Service) and so on.

3　The design of SBC on VOCAL

3.1　SBC deployment

A SBC is located at border of VOCAL system; every SBC added is a “door” for VOCAL system. VOCAL can have many "doors" (SBCs), and each SBC serves for a specified group of users, for example one SBC for one agent.

Only Marshal Server, the initial point of contact for all SIP signals that enter the VOCAL system, can be the connecting point between SBC and VOCAL system. Besides, one SBC can only map one Marshal Server, but one Marshal Server can map many SBCs. SBC deployment diagram is shown in figure 2.

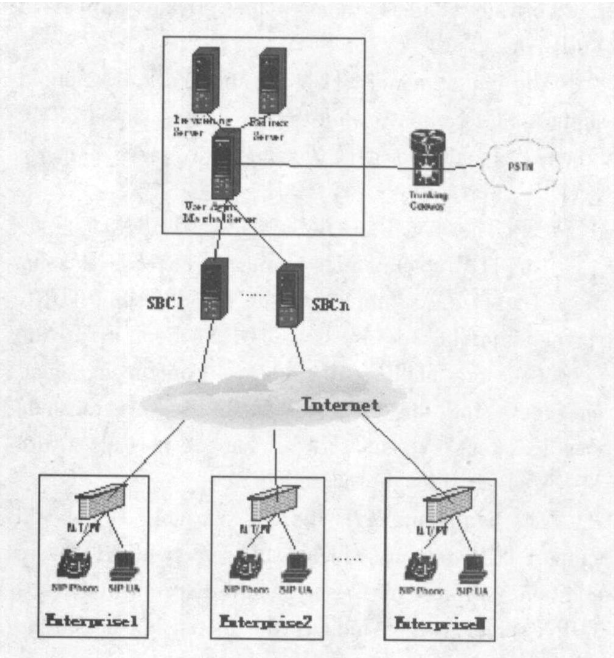

Fig. 2 SBC Deployment

3.2 Mechanism of MS

Before designing SBC, let's see the initial point of contact for all SIP signals that enter the VOCAL system, Marshal Server.

Figure 3 is the MS system. As the only interacts with SBC in SIP messages, we can set aside MS's communications with other servers and focus on how MS functions processing SIP message. At the bottom of MS, there is SIP stack, which achieves SIP messages packaging and transmission. SIPThread receives SIP messages and inserts them into SipProxyEvent queue; StatelessWorkerThread gets SIP events from SipProxyEvent queue and processes them, then sends them through SIP stack. For its non-state server model, the SIP events process is handled through the use of function operators. MS contains seven operators: MrshlOpAck, MrshlOpBye, MrshlOpCancel, MrshlOpInvite, MrshlOpOptions, MrshlOpRegister and MrshlOpStatus. The previous six operators process SIP request messages, and the last one process SIP status messages.

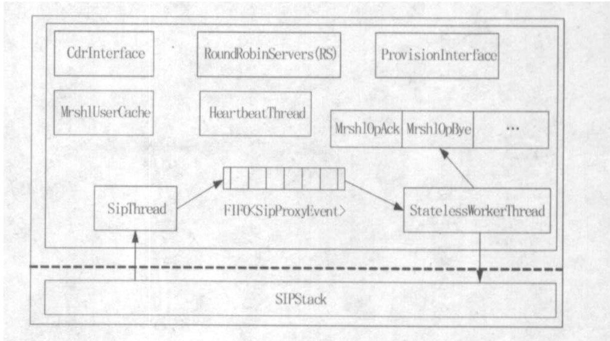

Fig. 3 Marshal Server System

3.3 Design of SBC

SBC itself can be seen as the proxy server support VoIP, which can handle SIP (the fifth level protocol) signals and manipulate the contents of these signals, thereby, achieving conversion from the private IP address to the public IP address; on the other hand, it could transit the media stream. Figure 4 is a chart of SBC.

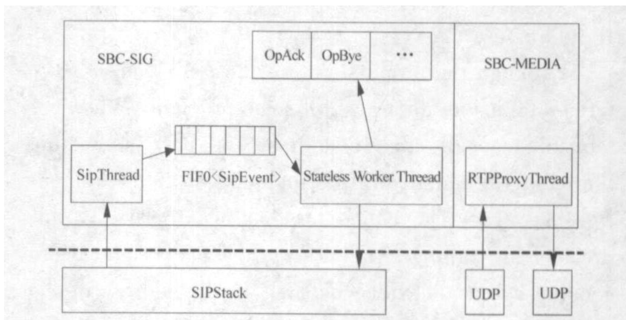

Fig. 4 SBC System

A SBC can be divided into two logically distinct modules: the Signaling SBC (SBC-SIG) and the Media SBC (SBC-MEDIA).

SBC-SIG processes SIP messages using the mechanism of MS. The difference between SBC-SIG and MS is the operator. StatelessWorkerThread processes SIP events by operators, and then sends them through SIP stack. The following section will discuss how SBC-SIG processes the SIP messages.

SBC-MEDIA allocates a RTP port for the transmission of media stream when a call is established. It starts up one thread for each call and the thread transmits the media packets of the call to the right IP address. SBC-MEDIA allocates a RTP port for the transmission of media stream when a call is established. After the session is established, SIP UA sends RTP packets every other time (such as 20s), so as to keep the same port in NAT and traverse NAT[6].

3.4 Message-processing in SBC

SBC functions can be summarized as the following: receiving data packets from outside and manipulating the contents of these messages before transmitted out. The exchange of SIP signals with the outside is a basic fixed mode, thus the key point is how to manipulate the contents of SIP signals received. There are two types packets received: SIP messages and media packets. A SIP message is either a request from a client to a server, or a response from a server to a client.[2] SIP requests include INVITE, ACK, BYE, CANCEL, OPTIONS and REGISTER; SIP responses include 1xx, 2xx, 3xx, 4xx, 5xx, 6xx.

3.4.1 REGISTER message processing SBC receives REGISTER message from user agent, and then put it into REGISTER message queue. REGISTER message processing thread will do with the message. The work that the thread must do is to:

* Get a REGISTER message from the queue; put the gateway address and "USER" from the content of the REGISTER message into user data storage table.

* Update the contact header field of the message with the address of SBC.

* Add the SBC address to the via header field of the message.

* Send the REGISTER message to Marshal Server.

* Delete the REGISTER message from the queue.

All the SIP messages from VOCAL system shall come across SBC, so we have to change the contact header field of REGISTER message with the SBC address which makes VOCAL locate users at SBC. SBC is able to look up the user data storage table, and send SIP messages to the corresponding user agent.

3.4.2 INVITE message processing SBC receives INVITE message and put it into INVITE message queue. INVITE message processing thread will do with the message. The work that the thread must do is to:

* Judge if the INVITE message comes from user agent or Marshal Server.

* Add the SBC address to the via header field of the message. If the INVITE message comes from user agent, we shall update the contact header field with gateway address.

* Allocate the RTP port for media stream of this call, and update the SDP content in INVITE message with SBC address and the RTP port.

* Put Call_Id of the INVITE message and RTP port allocated into Call_Id and RTP port storage table.

Other SIP requests BYE, ACK, CANCEL and OPTIONS have a similar processing with the INVITE message processing.

3.4.3 Response message processing SBC receives response message and put it into response message queue. Response message processing thread will do with the message. The transmission of response messages is obtained according to the via header field. SBC will send response messages according to the via header field. We must update SDP content in 200OK message responding INVITE message with the Call_Id and RTP port storage table, then send 200OK message according to the via header field.

3.4.4 Media message processing After a call established, both of user agents will send media packets basing on consultation addresses. The address, media packets will be sent to, is contained in the SDP content in INVITE message or 200OK message. SBC has updated the SDP content in INVITE message and 200OK message with its IP address and RTP port allocated, therefore all the media packets will be sent to SBC. SBC listens to the RTP port allocate to the call, if media packets come from user agent A, then SBC will send the media packets to the other user agent B, otherwise SBC will send the media packets to user agent A. When the session ended, SBC closes down the RTP port available for media stream.

4 Example

We construct a network to observe SBC processing IP messages. Figure 5 is the testing network architecture.

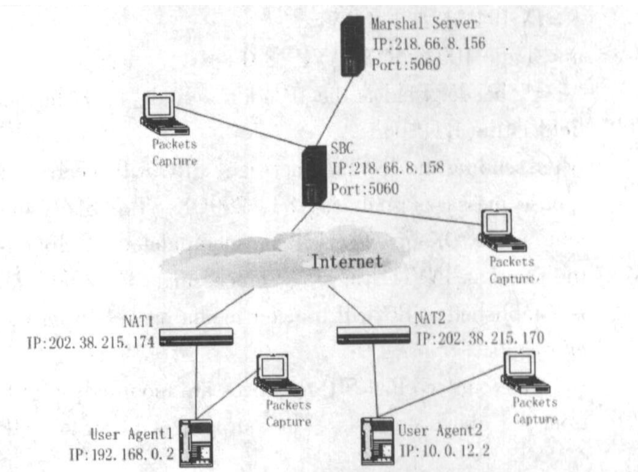

Fig. 5 Testing Network Architecture

The following is the packets analysis when User Agent 1 establishes a call with the public network user.

User Agent must register to VOCAL system. So UA sends REGISTER messages to SBC in order to access VOCAL. According to the REGISTER message processing, SBC update the contact header field of the REGISTER message with the address of SBC firstly.

The original contact header field is:

Contact: <sip: 11003@192.168.0.2:5080>

The contact header field updated is:

Contact: <sip: 11003@208. 66. 8. 158: 5080>

Second, add the SBC address to the via header field of the message.

The via header field added is:

Via: SIP/2. 0/UDP218. 66. 8. 158: 5080; branch = 4A5633B705545060

After succeeding in register, UA initiates a call. When SBC receives the INVITE message, firstly, SBC updates the contact header field of the message with the address of SBC.

The original contact header field is:

Contact: <sip: 11003@192. 168. 0. 2: 5080>

The contact header field updated is:

Contact: <sip: 11003@202. 66. 8. 158: 5080>

Second, add the SBC address to the via header field of the message.

The via header field added is:

Via: SIP/2. 0/UDP218. 66. 8. 158: 5080; branch = A7214FFD154D0F7F6

Third, allocate the RTP port for media stream of this call, and then update the SDP contents in INVITE message with SBC address and the RTP port.

The original SDP contents are:

c=IN IP4 192. 168. 0. 2

m=audio 9062 RTP/AVP 18 0 8 4

The SDP contents manipulated are:

c=IN IP4 218. 66. 8. 158

m=audio 10023 RTP/AVP 18 0 8 4

"c=" header field is the IP address and "m=" header field is the RTP port.

After sending the INVITE message, SBC will receive response messages such as 183, 200OK. The SDP contents in 200OK message will be manipulated by SBC as the same as INVITE message processing. Then the call is established. SBC will transfer media packets from User Agents.

We can see that SIP messages are modified by SBC from the above analysis. The purpose is to switch SIP messages through SBC so as to establish the call. And we allocate the RTP port for the call, so that RTP messages can be transmitted from the RTP port. Then, SBC for traversing NAT on VOCAL is implemented.

The thesis researches the SBC solution for VOCAL system traversing NAT/FW. VOCAL can traverse NAT/FW without any change with SBC. SBC has universal applicability and guidance for all the SIP-based VoIP systems traversing NAT/FW. SBC also could be extended. SBC can get more user information from SIP messages, like phone records. So users can query his phone records from SBC and agents can manage users. SBC also can measure and enforce QoS, and protect DoS attacks. Therefore, SBC is of great importance to extend VoIP services.

References

[1] HANDLEY M, SCHULZRINNE H, SCHOOLER E, et al. SIP: Session Initiation Protocol [S/OL]. March, 1999. RFC2543.

[2] Vovida. org. VOCAL 1. 5. 0 [CP/OL], http://www.vovida.org/applications/downloads/vocal/. April 1, 2003.

[3] TONG Xinzhe. VoIP Traverse NAT. Guangdong Shuju Tongxin, December, 2004.

[4] Lennox, Schulzrinne. CPL: A Language for User Control of Internet Telephony Services [S/OL]. March 10, 2000. draft-ietf-iptel-cpl-01.

[5] LIU Fagui, NING Jing. A General Method of New Feature Service Development on VOCAL [C]. Mobility Conference 2005.

[6] XIAO Shengren. Solution for Softswitch Based on SIP Protocol Traversing NAT [J]. Computer Knowledge and Technology, March, 2006.

[7] YAN Jun. Technology for NGN Network Traversing NAT [J]. Communication World, December, 2003.

[8] ZHU Shaohua. Research on Commerce NGN Core Net Work and Key Technology [J]. Guangdong Communication Technology, September, 2004.

基于 Vocal的私网穿透的 SBC（Session Border Controller）设计

刘发贵，饶　鹏

（华南理工大学计算机科学与工程学院，广东 广州 510640）

摘　要：随着技术的成熟，VoIP技术以更加迅猛的势头得到发展和普及，但是在这个过程中也不可避免的碰到一些困难和问题，VoIP应用如何穿越 NAT的问题就是其中之一。VOCAL是完全基于 SIP的 VoIP开放源代码方案。如何穿越 NAT也是用 VOCAL系统开发 VoIP服务必须解决的问题。研究了 NAT技术给 VoIP服务带来的问题，并且就当前主流的 NAT穿透方案进行探讨，然后在此基础上选择一个无须对现有设备进行任何改动的 VOCAL系统穿越 NAT方案——SBC。并以 VOCAL系统为对象，研究 SBC在 VOCAL系统中的设计。

关键词：IP电话；VOCAL；SIP协议；SBC；NAT