An Efficient NAT Traversal for SIP and Its Associated Media sessions

Yun-Shuai Yu, Ce-Kuen Shieh, *Wen-Shyang Hwang,
**Chien-Chan Hsu, **Che-Shiun Ho, **Ji-Feng Chiu
Department of Electrical Engineering, National Cheng Kung University, Taiwan R.O.C.
**Department of Electrical Engineering, National Kaohsiung University of Applied Sciences, Taiwan R.O.C*
***Computer & Communications Research Laboratories, Industrial Technology Research Institute, Taiwan R.O.C*
kaede@hpds.ee.ncku.edu.tw, shieh@ee.ncku.edu.tw, wshwang@mail.ee.kuas.edu.tw
hcc@itri.org.tw, hocs@itri.org.tw, jfeng@itri.org.tw

Abstract-*Session Initiation Protocol (SIP) standardized by IETF provides a way to establish the sessions between Internet telephony devices, but this protocol can't work with Network Address Translator (NAT). Therefore how to make SIP NAT-friendly becomes a worthy topic. This paper proposes an efficient approach TAB (Triggering Address Bindings on NAT), which helps SIP devices establish the shortest paths for the media sessions without modifying the existing NAT products. The approach is validated, and the results show that the traffic delay time between devices with TAB is reduced substantially than that with other solutions.*

Keywords: SIP, NAT, TAB.

1. Introduction

Session Initiation Protocol [1] is a signaling protocol for establishing, modifying and terminating multimedia sessions such as Internet telephony calls with one or more participants. SIP defines two basic classes of network entities: clients and servers. In the case of IP telephony, the caller acts as the client and the callee acts as the server. Generally, the caller does not know the callee's address, so the caller can't set up sessions with the callee directly. For locating prospective session participants, a special server, named Registrar, is designed to receive the registrations about the current locations of end devices, as shown in Figure 1. Moreover, a proxy is placed in the signaling path between end devices. When the proxy receives a SIP request, for example an Invite message, it would query the registrar about the location information of the callee and then route the request on toward its destination. In most cases, the registrar and the proxy are implemented on a single device. The message body of the request provides a description of the session to be established. The description contains media type, codec type, and the address for receiving data, etc. It is used for negotiation with the called party. The callee processes the request and next generates a response to notify the caller whether it accepts the call or not. The callee also puts some information about the expected session in the message body of the response. If both of them agree with the demands of each other, a media session can be established.

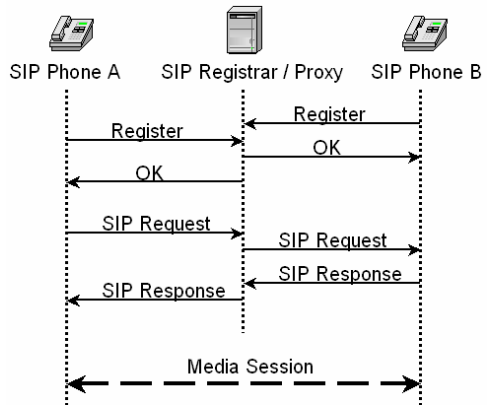

Figure 1. Overview of SIP operation

The above operations function well in the public network but fail when traversing through the NAT [2]. NAT was developed to allow the use of a single IP address for a whole network of computers to alleviate the problem of IP addresses shortage. Therefore NAT products are popular in today's world. Although providing many benefits, NAT also incur some problems. NAT only permits the devices in the private realm to initiate the sessions, so devices behind a NAT may be able to make SIP calls but not be able to receive them. Furthermore, the private IP addresses inserted by client devices in the packet payload are not routable in public networks, so the address information carried in the SIP messages becomes invalid after passing through a NAT. The media sessions can't be established if the client devices use these private addresses to send or receive media. In order for SIP itself and its associated media sessions to traverse NATs, this paper proposes an approach TAB to solve these problems. TAB makes

use of the NAT characteristic by *triggering address bindings on the NAT and keeping these bindings alive for the traversal of SIP messages and media traffic, so there is no need to modify the NAT.* In this paper, the devices with TAB in some network configurations are studied to understand how to properly establish communications in the NAT environment. Moreover, the paper will focus on media streams that are carried over the Real- time Transport Protocol (RTP) [3]. In all cases, only RTP is shown and discussed, to simplify the discussion. The related RTCP sessions can be established in the same way.

This paper is organized as follows. Section 2 describes the related study. The approach of TAB is described in section 3. Section 4 describes the implementation of TAB and discusses the experimental results. A conclusion and future work is given in the last section.

2. Related Work

There are some solutions without changing the design of NAT for allowing SIP messages or its associated media sessions to pass through NAT. For SIP traverses NAT, RFC 3581 [4] defines a SIP extension for symmetric response routing when SIP request operates over User Datagram Protocol (UDP). For media sessions traverse NAT, a lightweight detection protocol, STUN [5], allows a device inside the NAT to determine the NAT's behavior and bindings indirectly, and to modify the protocol messages appropriately. The defect of STUN is that it can't work with symmetric NAT, which is widely used in today's enterprise. Therefore STUN can't provide a complete solution. TURN [6] solves this problem by relaying data through a server that resides on the public Internet. A device behind the NAT would use TURN protocol to get the address and port on the TURN server and then use them to invite its peer. It is a feasible way to pass through all kinds of NAT, but it also comes at high cost to the provider of the TURN server. Thus it is recommendable to use TURN as the last recourse, preferring other mechanisms (such as STUN or direct connectivity) when possible. To accomplish this, ICE [7] methodology can be used to discover the optimal means of connectivity. However there is an exception; if one party is behind a symmetric NAT and the other is not, ICE still chooses TURN, which is not the shortest way. In this paper, a Triggering method, called TAB, is studied to seek or create the shortest path for SIP messages and its associated media sessions to pass through NATs.

3. TAB Approach

TAB is a complete solution of NAT traversal for SIP. As mentioned earlier, SIP has to solve two problems: the first one is to initiate sessions from public networks into a private network, and the other is to modify the address information in the SIP messages into a reachable one. The details about how TAB achieves these goals are described as follows.

3.1. Solutions of NAT traversal for SIP itself

The end device behind a NAT can transmit a SIP request to the public network. While the request is carried by UDP, RFC 3581 provides a way that allows the SIP devices to send responses back through the NAT. However, few products support this method now. TAB suggests the end devices to use Transport Control Protocol (TCP) to send and receive SIP messages. The address binding on the NAT for the TCP connection will be kept alive until a TCP FIN packet is sent. The server can respond the client via the same TCP connection without increasing the operational complexity.

It is impossible for a device to originate a session from public network to its peer that locates in a private realm except when it knows the externally assigned IP address ahead of time. A proprietary server, named TAB server, is designed to solve this problem. A TAB server is an entity that can tell the client whether it is behind a NAT. It can also act as a relay, receiving packets on the address it provides to clients, and forwarding them to the clients. The operation is illustrated in Figure 2. Before registration, the client, SIP phone A, sends a TAB request (message F1) using UDP to the TAB server. The application payload of the request contains the address information about where it comes from. TAB server compares this information with that in its IP and UDP header. If they are different, it means there is at least one NAT in the path between the client and the TAB server. Then TAB server allocates a public transport address named SRA (SIP Relay Address) for relaying SIP messages to the client in the future. Afterward a TAB response is sent to the client (message F3); the response includes SRA and the network configuration. After the client knows it is in a private realm, it will register its location using SRA, instead of its local address (message F5).

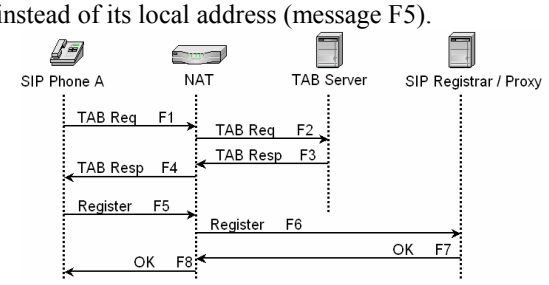

Figure 2. Registered with the public transport address provided by TAB server

Hereafter, SIP proxy will deliver SIP requests to SIP phone A's SRA (Message F9 in Figure 3). When TAB server receives packets on the SRA, it relays them towards SIP phone A (Message F10). Since the TAB request has triggered an address binding on the NAT, and TAB server will periodically send dummy packets to keep the binding alive, the relayed packets can pass through the NAT and finally reach SIP phone A. Afterward, SIP phone A responds SIP responses to the TAB Server, and TAB server will forward these SIP messages to the SIP proxy (See Message F12, F13, and F14).

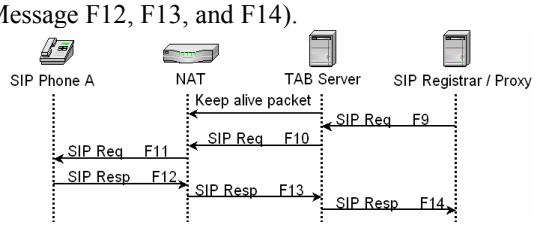

Figure 3. Session originated outside the NAT

3.2. Solutions of NAT traversal for media sessions

If the client wishes to establish a media session with the peer, it needs to do some additional procedures as follows before it makes a call initiation request (INVITE message) or a success response (200 OK message). First, it transmits a TAB request from the port, which will be used for RTP communication with the peer. The content of the TAB request is the local address where it originates. TAB server will allocate a public transport address named MRA (Media Relay Address) to relay the media traffic for the client. A TAB response containing the information of MRA is returned to the client. Then a description of the media session in the SIP message body is written in Session Description Protocol (SDP) [8] to inform the peer where the client expects to receive the RTP traffic on.

```
v=0
o=jack 2691444900 2487311000 IN IP4
host.tab.com
s=test
c=IN IP4 140.116.72.98
t=0 0
m=audio 32354 RTP/AVP 0
a=local_addr:192.168.0.1 1200
```

Figure 4. The SDP text message offered by the client behind a NAT

Figure 4 is an example of a SDP description. The SDP text message includes protocol version ("v="), owner and session identifier ("o="), session name ("s="), connection information ("c="), time interval the session is active ("t="), media name and transport address ("m="). MRA is put on the "m=" and "c=" fields, since this is the address that can work with the traditional SIP device. A new attribute is defined as local_addr in the "a=" field to show the local transport address of the client's application. If the peer also supports TAB, additional procedures will be taken after the exchange of SIP messages. In Figure 5, SIP phone A is behind a NAT but SIP phone B isn't. Hence B can transmit RTP traffic to A's MRA, and then TAB server will relay the packets to A. This path is not the shortest, so A sends a Trigger Packet to B for triggering an address binding on the NAT. Now, B is aware of the shortest path, so it will send a message to inform TAB server to close A's MRA. Finally, B sends all the traffic to the source transport address of the Trigger Packet.

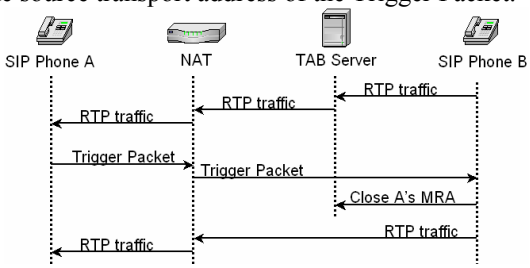

Figure 5. Trigger the shortest path

When both of the participants are behind NAT(s), they need to do a Connectivity Check further to verify whether they are in the same private realm. The Connectivity Check is an UDP packet sent form one's local transport address to the other one's. If the check is success, they can communicate with each other directly (See Figure 6). If not, they will send traffic to the MRA of the other party for relay (See Figure 7).

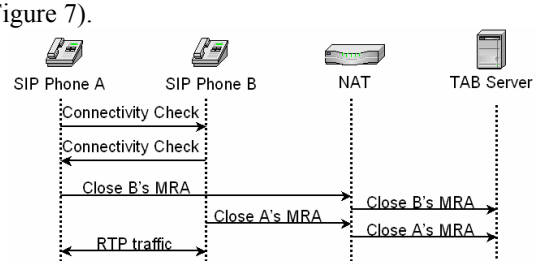

Figure 6. Behind the same NAT

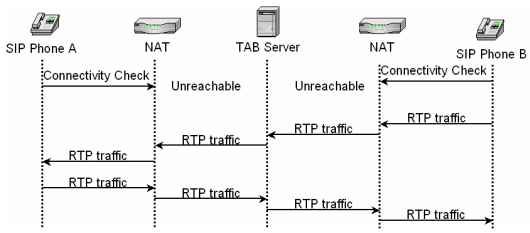

Figure 7. Behind different NATs

For security concern, it is important that the Trigger Packet and the Connectivity Check packet should be authenticated in some way.

What needs to be added to a SIP client to support TAB is summarized as follows. First, the client has to send a TAB request to get a SRA. If the client is in a private network, it should register its address with the SIP proxy using the SRA. Before inviting or answering its peer, it sends another TAB request to

get a MRA. Then it inserts the MRA and its local address into the SIP message. After exchanging the SIP messages, the SIP client knows whether its peer is behind a NAT or not. If the peer was in the public network, the SIP client will use Trigger Packet to establish the shortest communication channel. If not, it will do a Connectivity Check.

4. Implementation and Experiments

To verify our TAB approach, we construct a SIP-based experimental platform, as shown in Figure 8. The basic specifications of the devices are listed in Table 1, and those devices are linked with 100-Mbps Ethernet connections. This platform comprises three main components, namely, SIP phones, a SIP proxy, and a TAB server.

SIP Phone: A SIP phone is an entity to create and receive SIP requests and responses. These SIP phones can perform TAB functions.

SIP proxy: It includes a registrar server and a stateful proxy server. The registrar server receives updates from users about their current addresses, and the proxy server is responsible for routing SIP messages.

TAB Server: It is implemented to detect whether the client is behind a NAT, and allocate public transport addresses for SRA and MRA, if needed.

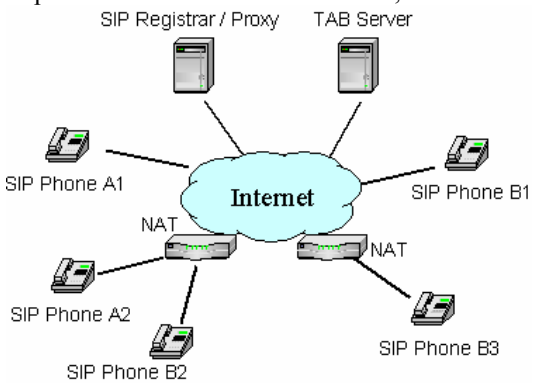

Figure 8. Experimental platform

Table 1 The basic specifications of the devices

	O.S.	CPU	RAM
TAB Server	Linux	2.8G	256M
SIP Proxy	Win2000	850M	256M
SIP Phone	Win2000	366M	192M
NAT	Linux	300M	128M

Currently, the SIP phones and the SIP proxy make use of NIST-SIP 1.2 [9] for testing. NIST-SIP is a distribution containing a SIP protocol stack/library that will help people build SIP applications and servers. It is a reference implementation for JAIN-SIP 1.1 [10], a low level protocol API for SIP. Some lightweight procedures are incorporated into SIP phones for TAB enabling. In addition, the TAB server is coded by C language.

The first experiment estimates how long the TAB approach needs to initiate a call. For traditional SIP client, the call initiation time is from the time it registered with the SIP proxy to the time it got a SIP response (200 OK) to the call invitation. For the SIP client which supports TAB, it is from the time the client generated the TAB request to get a SRA to the time the client completed the Trigger procedure or Connectivity Check. In Fig. 8, five scenarios are tested. (1) Both the caller and the callee are in the public network (B1 calls A1). (2) The caller is in the public network, while the callee isn't (B1 calls A2). (3) The caller is behind a NAT, while the callee isn't (A2 calls B1). (4) The two devices are behind the same NAT (A2 calls B2). (5) The two devices are behind different NATs (A2 calls B3). Traditional SIP devices can only work in scenario 1, while TAB devices can work in all scenarios. The experimental results are shown in Figure 9. *The overhead of TAB for initiating a call is about 40 to 150 ms, which can be negligible for human beings.*

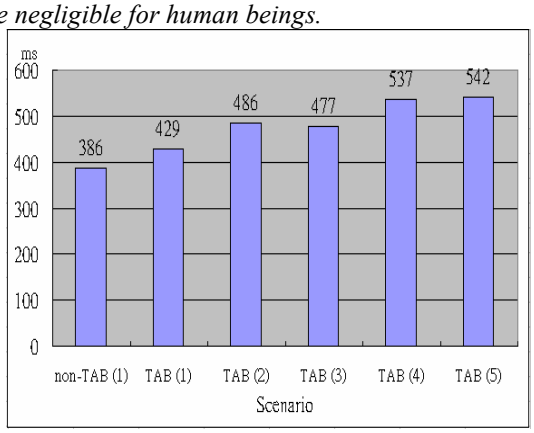

Figure 9. Call initiation time

Second, in scenario 2 and 3, if the NAT is symmetric, TURN or ICE will choose to relay the media traffic. TAB performs better than these two mechanisms because it uses a Trigger Packet to trigger an address binding on NAT for establishing the shortest path. Traffic delay time is from the time SIP phone B1 sent out a 150-byte RTP packet to the time SIP phone B2 received the packet. The performance of those solutions is rated by measuring the traffic delay time. The time needed for relay is 1499(us), while the time needed for TAB is 1210(us). *About 20% traffic delay time was reduced. TAB approach will outperform TURN approach when the load of the server or the network became heavy.*

5. Conclusion and Future Works

In this paper, we have proposed a complete solution for establishing sessions using SIP through NAT. An efficient approach called TAB was designed for implementing at both the end devices and the TAB server. TAB did not modify the design of NAT, so there was no need to upgrade the existing NAT products. Moreover, TAB was workable for all

kinds of NATs and compatible with traditional SIP devices located in public network. Despite the network topology and deployment configuration, TAB can find the shortest paths for the media sessions with only a trivial overhead. We plan to incorporate the TAB server into the SIP proxy server to reduce the overhead. More concern will be put on the security in the future.

References

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

[2] P. Srisuresh, K. Egevang, “Traditional IP Network Address Translator (Traditional NAT)”, RFC 3022, January 2001.

[3] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

[4] J. Rosenberg and H. Schulzrinne, “An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing”, RFC 3581, August 2003.

[5] J. Rosenberg, J. Weinberger, C. Huitema and R. Mahy, “STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)”, RFC 3489, March 2003.

[6] J. Rosenberg, R. Mahy and C. Huitema, “Traversal Using Relay NAT (TURN)”, draft-rosenberg-midcom-turn-03, October 2003.

[7] J. Rosenberg and G. Camarillo, “Examples of Network Address Translation (NAT) and Firewall Traversal for the Session Initiation Protocol (SIP)”, draft-rosenberg-sipping-nat-scenarios-02, December 2003

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

[9] About the IP telephone project. http://snad.ncsl.nist .gov/proj/iptel/. 2004

[10] JSR 32: JAINTM SIP API Specification. http://www.jcp.org/en/jsr/detail?id=32. August 2003.