CDCS: a New Case-Based Method for Transparent NAT Traversals of the SIP Protocol

Mustapha GUEZOURI
LISSI/SCTIC, University of Paris XII-Val de Marne, France
e-mail mguezouri@yahoo.fr
and
Abdelhamid MELLOUK
LISSI/SCTIC, University of Paris XII-Val de Marne, France

ABSTRACT

Voice communications on IP networks use owner protocols as well as standards like SIP, MGCP and H323. In this paper we propose a new method for transparent traversal of NATed (Network Address Translated) networks for the SIP (Session Initiation Protocol) protocol. Although SIP is an application layer protocol, its operation is affected by address translation. This is because SIP uses network layer information (source IP and source port) that is lost by the NAT operation.

The suggested method adapts dynamically one of the three solutions: Connection-Oriented media STUN or TURN depending on the situation occurring during call initiation.

Key-words:
SIP, NAT, STUN, TURN, ICE, Connection-Oriented media, Voice over IP, IP Telephony.

1. INTRODUCTION

SIP standard (Session Initiation Protocol) [1] was in gestation for many years and implemented only in companies' platforms (routers, telephones, etc.) for applications such as IP Telephony and videoconference. Now, more and more service providers and carriers integrate SIP in their commercial offers and it seems that SIP is well positioned in the market.

SIP is an application layer protocol very sensitive to Network Address Translators (NAT) [2] [3]. This, because it uses network layer information that is lost when translated. NAT traversals cause two major problems in SIP operation. The first occurs in the signalling stage, while the second takes place during the multimedia session.

While the first problem can be overcame using SIP protocol extensions [5] or TCP (Transmission Control Protocol) [6] connections. The latter is quite difficult because of the information description in the body of the *INVITE* request and the corresponding response *OK:200* in the SDP stage (Session Description Protocol) [7] (UDP ports for each client have local significance only and are invisible from the outside).

Many solutions to theses issues were suggested: TURN (Traversal Using Relay NAT) [8], STUN (Simple Traversal of UDP through NAT) [9], Connection-Oriented media [10] and ICE (Interactive Connectivity Establishment) [11]. But each of them presents a number of weaknesses depending on the configuration in use.

This paper introduces a new case-based method for call setup (CDCS) for the SIP protocol. Sections 2 through 5 review the TURN, STUN, Connection-Oriented media and ICE methodologies. Sections 5 through 7 details CDCS and discuss implementation issues.

2. TURN PROTOCOL

TURN protocol allows units behind firewalls or network address translators to communicate through TCP connections or UDP (User Datagram Protocol) [12]. The key idea in TURN operation is very simple; each unit whishing to communicate, reserves a public IP address along with a number of needed ports. This process is independent of the call initiation and hence, is resource consuming.

3. STUN PROTOCOL

STUN protocol is used by the communicating units to detect the presence of NAT and their corresponding public IPs addresses and ports numbers. During the process of call setup, clients use the detected information (if any) in the SDP session, making it possible for their peers to reach them. In case of a symmetric NAT, call setup is impossible using the STUN protocol.

4. CONNECTION-ORIENTED MEDIA

Connection-Oriented media provides a solution to the problem of session description. This technique is used to establish a multimedia session between two clients if one of them is not behind a NAT. The key idea here is that the client behind the NAT should initiate the session so that the other client could determine the IP address and port number for the RTP/RTCP (Real-Time Transport Protocol/ Real-Time Transport Control Protocol) [13] packets.

5. ICE METHODOLOGY

ICE (Interactive Connectivity Establishment) considers that clients could be joined through multiple IP addresses and port numbers at the same time. Therefore, and before call setup, a client determines all the combinations of IP addresses and port numbers through which it could be joined using protocols such STUN and TURN. Then it places all the found combinations in the description of the multimedia session. The client on the other side of the call receives the call setup and starts a connectivity control process for all the IP addresses and port numbers found in the SDP body of the request in the same way as the caller. At the end both clients know the public IP address and port number of each other and RTP/RTCP packets could be exchanged. This method generates a considerable amount of messages, slowing down the call setup.
Table 1 summaries the methods used in today's environments and their weaknesses.

Table 1: Known weaknesses of TURN, STUN, Connection-Oriented Media and ICE methods

Method	Related weaknesses
TURN	Resource consuming
STUN	Does not work with symmetric NAT
Connection-Oriented Media	Works if only one of clients is behind a NAT
ICE	Generates a significant amount of messages, even though both clients are not behind a NAT

6. CDCS METHOD

Case Driven Call Setup method (CDCS) is a new method that applies the adequate solution (STUN, TURN, connection-oriented) for call setup depending on the current configuration. CDCS goes through two distinct phases. First, it detects the presence of NAT and its type. This is possible by using the STUN protocol. Then the client gets the Proxy server informed of the result by sending it a new field "*NAT-Type*" in the *REGISTER* request. *NAT-Type* field takes the following values depending on the detected NAT: *no-NAT, full-cone-NAT,* and *other-NAT*. The proxy server then, authenticates the client and stores its external IP address as well as its local IP address contained in the *Via* field and the type of NAT in use (figure 1). According to the configuration, the proxy server chooses the adequate solution.

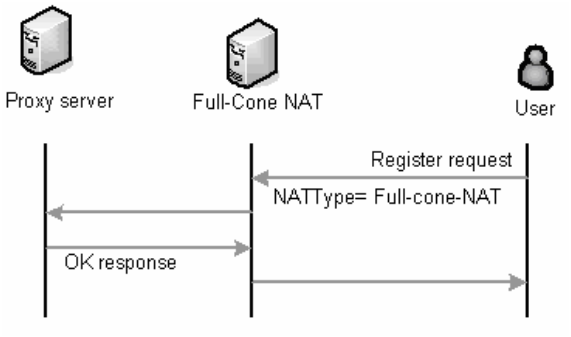

Figure 1: Client registration example

6.1 Call setup procedure

To establish a call, the originating client sends an *INVITE* request and listens for STUN primitives on each port specified in the SDP body. Upon the receipt of the *INVITE* request, the proxy server queries its localization service or database to determine both the caller and callee locations and select the appropriate solution.
CDCS deals with all possible configurations. The simplest one is that, where both the caller and the callee are not behind any NAT. This case is not problematic and needs no treatment. In the following, the other configurations are discussed

6.1.1 Case 1: Either the caller or the callee is behind a NAT

The appropriate method in this case is Connection-Oriented media. If the caller is the one behind the NAT (figure 2), the proxy server adds *"a=active direction"* in the SDP body of the *INVITE* request. This tells the callee to not send its RTP/RTCP [10] packets before receiving the RTP/RTCP packets of the caller. Otherwise, if the callee is the one behind the NAT, the proxy server sends an *INVITE* request with *"a=passive direction"* in the SDP body. This way, the caller waits for the RTP/RTCP packets of the callee in order to determine the destination IP address and port number of the RTP/RTCP response packets.

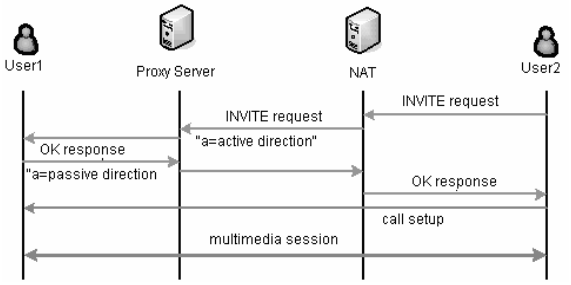

Figure 2: Multimedia session establishment for case 1: caller behind NAT

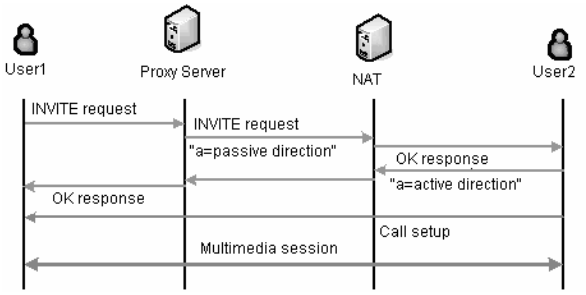

Figure 3: Multimedia session establishment for case 1: callee behind NAT

6.1.2 Case 2: Caller behind a full-cone NAT and callee behind a non full-cone NAT (figure 4)

Upon the receipt of the *INVITE* request, the proxy server tells the caller to retransmit a new *INVITE* message with the appropriate NAT mappings in the SDP body. This is possible by using the STUN protocol. Before, the caller sends a *"Discovery RTP request"* and a *"Discovery RTCP request"* to create the NAT entries for RTP and RTCP on the full-cone NAT machine. Finally, the Connection-Oriented media is adapted to setup up a multimedia session between the two clients.

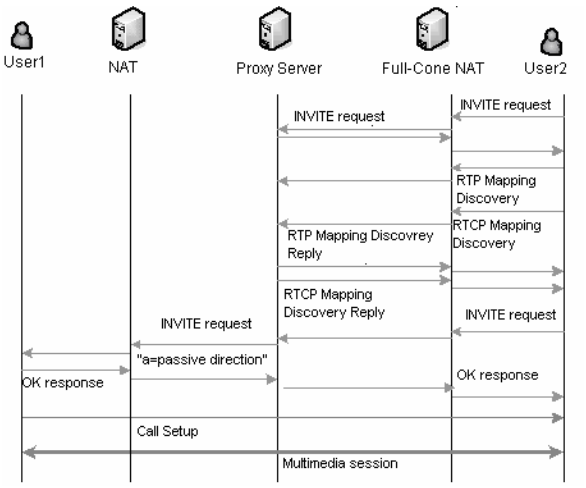

Figure 4: Multimedia session establishment case 2

6.1.3 Case 3: Callee behind a full-cone NAT and caller behind a non full-cone NAT (figure 5)

In this case, the proxy server uses a new field "*process-STUN"* in the *INVITE* message to tell the callee to write down the appropriate NAT mappings in the SDP body and do the necessary to create the NAT entries for the RTP/RTCP traffic on the full-cone NAT.

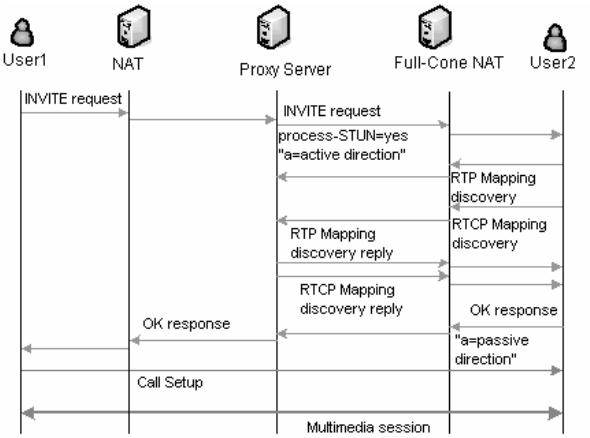

Figure 5: Multimedia session establishment case 3

6.1.4 Case 4: Caller and callee behind symmetric NATs (see figure 7)

In this case, the proxy server uses the same procedures as the TURN protocol. It allocates the necessary resources, IP addresses and port numbers for the clients and moreover it becomes an intermediate node during the *RTP/RTCP* exchange.

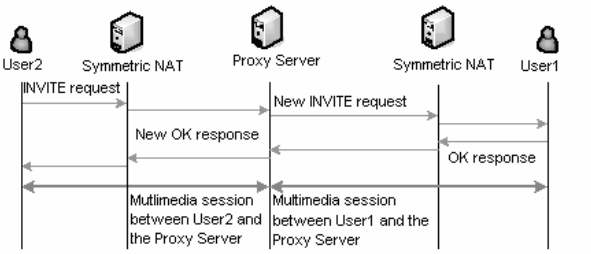

Figure 6: Multimedia session establishment case 4

6.1.5 Case 5: Caller and callee on the same network/sub-network

This case is identified if both caller and callee have the same public IP address. The proxy server allocates then, the necessary resources for the two clients and adds *"a=alt 1: 1.0: Caller-username Caller-password Caller-local-IP-address Port''* [14] line in the SDP body of the *INVITE* request. This tells the callee to first, attempt to join the caller by its local address and if this fails, use the allocated IP addresses and port numbers contained in the "m" and "c" lines.

Note that if the local addresses are used, the proxy server frees the allocated resources for a further use.

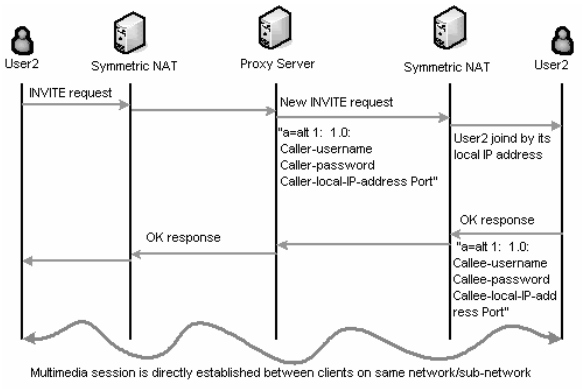

Figure 7: Multimedia session establishment case 5

7. CDCS ALGORITHM

To identify the occurring case, CDCS applies a series of tests (table 2) in the order shown in the flow diagram of the figure 8. The caller (respectively callee) is behind a NAT of type NATType1 (respectively NATType2) and takes a public IP address ExtAdd1 (respectively ExtAdd2) when translated.

Table 2: CDCS Algorithm Tests

Test	Description
Test 1	NATtype1=*no-NAT* AND NATtype2=*no-NAT*
Test 2	NATtype1=*no-NAT* XOR NATtype2 =*no-NAT*
Test 3	NATtype1=*full-cone-NAT* AND NATtype2=*other-NAT*
Test 4	NATtype1=*other-NAT* AND NATtype2= *full-cone-NAT*
Test 5	AddExt1=AddExt2

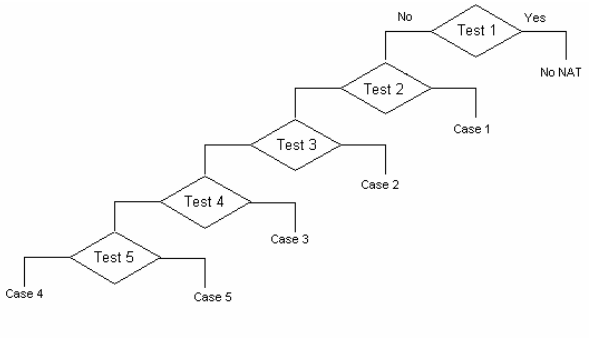

Figure 8: Case identification flow diagram

8. TESTS AND RESULTS

The TURN, STUN, Connection-Oriented media, ICE and the new CDCS solutions were tested for all the cases discussed in the previous sections on a platform of type PC-to-PC running a very simple home made soft IP phone that integrates the new SIP features. The different NAT types were implemented by adapting the *iptables* tool under the Linux kernel 2.4.18 to meet the desired NAT behaviors. We also developed a very small proxy server compliant with the CDCS requirements. Table 3 summarizes the obtained results.

Table 3: Reference Information about test results

	TURN	STUN	Connection-Oriented Media	ICE	CDCS
Case 1	yes[1]	yes[2]	yes[2]	yes[3]	yes
Case 2	yes[1]	yes[2]	no	yes[3]	yes
Case 3	yes[1]	yes[2]	no	yes[3]	yes
Case 4	yes[1]	no	no	yes[3]	yes
Case 5	yes[1]	no	no	yes[3]	yes

Remarks:

[1] For each communication, TURN allocates a public IP address and the multimedia session crosses the proxy server.

[2] The proxy server doesn't allocate any public IP for the communicating units.

[3] Generates a huge amount of messages: in case of 1 public IP address used for NATing each client, the number of sent messages for the *INVITE* request and its corresponding *OK:200* response is (65536-1024)*2= 129024

9. CONCLUSION

Unlike the other suggested solutions, the new CDCS method provides a transparent traversal of NATs for the SIP protocol. The undertaken experiments show that CDCS runs for all the possible configurations that may exist. Moreover, CDCS saves resources and adapts in a dynamic way the appropriate call setup for each identified case.

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] Egevang, K. and P. Francis, "The IP Network Address Translator", RFC 1631, may 1994.

[3] Srisuresh, P. and M. Holdreg, "IP Network Address Translator (NAT) Terminology

and Considerations", RFC 2663, august 1999.

[4] Rosenberg, J., D. Drew, and H. Schulzrinne, ‘’Getting SIP through Firewalls and NATs’’ *Internet Draft, Internet Engineering Task Force*, February. 2000. Work in progress.

[5] J.Rosenberg, H. Schulzrinne, and J. Weinberger. "An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing" Internet Draft, Internet Engineering Task Force, 27 September 2002. In progress Work.

[6] J. Postel, "Transmission Control Protocol", RFC 793, September 1981.

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

[8] J. Rosenberg, J. Weinberger, R. Mahy, and C. Huitema, "Traversal Using Relay NAT (TURN)", *Internet Draft, Internet Engineering Task Force*, 3 March 2003.

[9] 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.

[10] D. Yon "Connection-Oriented media Transport in SDP", Internet Draft, Internet Engineering Task Force, March 2003.

[11] J. Rosenberg. "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for the Session Initiation Protocol (SIP)" Internet Draft, Internet engineering Task Force, 24 February 2003.

[12] J. Postel, "User Datagram Protocol", RFC 768, September 1980.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[14] J. Rosenberg. "The Alternative Semantics for the Session Description Protocol: Grouping Framework", *Internet Draft, Internet Engineering Task Force*, February 2003.

[15] A. Mellouk, M. Guezouri"A new methodology to adapt SIP protocol for voice traffic transported over IP Network AICT/ICIW February 2006, Guadeloupe, French Caribbean. IEEE Computer Society 2006.

Received: Jul. 2006. Accepted: Jan. 2007.