VoIP P2P-SIP System

With NAT Traversal

Supervisor:
Prof. Lea. C. T.

Ho Yeung, Lee
Project Overview

Introduction

There are many solutions to traverse NAT, they are difficult to program and integrated in purely peer-to-peer software. The reasons for these difficulties are as follows. First, there is a third client or server act as a signaling server to do initial exchange for peers getting internet address and other information of each side. Second, routers block unauthorized access. There may be many cases when considering Transmission Control Protocol and User Datagram Protocol. Third, it includes the steps of processing at least one situation that is necessary to use relay.

Aim

The aim of this report is to introduce a design framework for Voice-Over-IP with Network Address Translators traversal.

There are 4 types of NAT, Cone NAT, Address Restricted NAT, Port Restricted NAT and Symmetric NAT. According to the approach, the first three can be combined as non-symmetric type. Finally, they are divided into two types, non-symmetric and symmetric. Non-symmetric is the general case solved by Hole Punching. Symmetric is the enterprise case solved by Traversal Using Relay NAT in both TCP and UDP.

<table>
<thead>
<tr>
<th></th>
<th>UDP</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Symmetric</td>
<td>UDP Hole Punching</td>
<td>TCP Hole Punching</td>
</tr>
<tr>
<td>Symmetric</td>
<td>Traversal Using Relay</td>
<td>Traversal Using Relay</td>
</tr>
</tbody>
</table>

Table 1 – NAT Traversal Solutions
Methodology

TCP & UDP Hole Punching:
Firstly, Clients do initial exchange with login server. Then, clients start listening and opened connection to each other nearly at the same time. Once one side mapping is created, another side can pass through the NAT with TCP. Either one or both socket is successful created, and then the socket will be used to send and receive data.

Figure 1 – Punching

Figure 1 – Punching Process
Traversing Using Relay NAT:
Assume that client A connects to port 6677 and ask the server to allocate a port for it, say, port 5678. Server returns the mapped address (server’s IP and port 5678) to client A. Client A sends the destination IP address, which it wants to talk to client B, to the server, then server adds the permission for client B to connect to port 5678 which has been allocated for client A. Client B wants to talk to client A, it has to get client A’s mapped address through a signaling server. Client B sends data to server’s port 5678, server relays the data to client A; client A sends data to server’s port 6677, server relays data to client via port 5678.

Figure 3 – TURN Process

Results
TCP and UDP are successfully connected and do two way communications.