Development of Traversal Using Relay NAT (TURN)

Supervisor:
Dr. Danny Tsang

Students’ Names:
Wong Yin Fai Kevin
Tang Ka Yi
Chan Chun Phang
Introduction

The well-known VoIP software, called SKYPE is very popular among youngsters because of its good quality in voice transmission. It is also famous for its reliability in establishing connection between clients successfully regardless of distances. It is an important feature that the application could traverse through NAT (Network Address Translation) smoothly. In recent years, VoIP services have tended to be deployed in Web based application. As a result, a complete solution to solve the problems brought from NAT should be developed. There is number of solutions implemented which could solve the problems to certain extents but not in all cases. Traversal Using Relay NAT (TURN) is a new protocol which could traverse through NAT successfully including symmetric NAT on both UDP and TCP connection.

Aim and Objectives

Simple Traversal of UDP through Network Address Translation (STUN) is one of the solutions that can solve the address translation problem. However, STUN fails to traverse through a symmetric NAT. In order to solve the problems perfectly, we would like to develop:

- **TURN** Server Protocol
- **TURN** Client Protocol

![Figure 1: System diagram](image)
Methodology

Client:
Figure 2 is a brief description of the actions taken by the TURN client. A client first sends an Allocate request, and then the TURN server sends appropriate responses back to the client. These responses are processed by the client as shown in the figure. If the response is an Allocate Response, the client will proceed to send a Send Request. If it is a Send Response, the client can start to send UDP data packet. If it is a DataIndication, the client will get the data encapsulated in it. If it is an Error Response, another Request with the same type will be sent again.

Server:
Figure 3 is a brief description of the actions taken by the TURN server upon reception of different requests. These requests are processed by the server as shown in the figure. If the request is an Allocate Request, the server will proceed to send an Allocate Response. If it is a Send Request, the server will proceed to send a Send Response. If it is an UDP data packet, the server will get the data and encapsulated it into a DataIndication message. If the requests are invalid, responses with the same type will be sent.
From figure 4, the TURN client A with IP address 192.168.1.102 which behind NAT could be allocated an address 143.89.192.43 and port 1111 from TURN server for receiving incoming connection in later time.

Figure 4: Allocate Response

From figure 5, client A informed TURN server that he would like to communicate with client B (IP address 143.89.80.47) including the message (how are you). The request was successfully received by the TURN server and the same data was retrieved.

Figure 5: Send Request

From figure 6, the TURN server was succeed in relaying the message “how are you” to client B.

Figure 6: Data Indication