SIP SPEAKER

BY SIKE HUANG AND SHANBO LI

1. PROBLEM FACED & SOLUTIONS

1.1 IMPLEMENTATION OF SIP PROTOCOL

Since we are group of two persons, we install SIP phone software on our own laptops, and SJphone and Express Talk are ones we decide to use. One of us calls another, and WireShark is utilized to capture the SIP traffic in order to analysis the package format, and we also check the RFC to understand the meaning of such format. It is good that the SIP/SDP are merely textual based protocol, therefore they are easy to understand. So in our SipSpeak application, we basically just send lines of text by UDP datagram according to the format redefined.

1.2 REAL-TIME TRANSMISSION OF VOICE

This part is totally taken care of by Java Media Framework. We look at the tutorial provided on Sun's website and use the example code. However there are few errors in the sample code and it took us very long time to fix it. The basic idea is that, after SipServer establishes the SIP handshake with clients, it asks RtpTransmit to launch the RTP session with specific client.

1.3 DYNAMICAL GENERATION OF SOUND FILE

The Web server coded in the assignment is used as the front user interface, where user can change the message that will be played from server side. Lot of refractors has been made to the web server, we'd rather say that it becomes much simpler http server comparing to previous one. The duty of converting text to speech is totally covered by FreeTTS library, we also study the tutorial offered on its official website, the library is very handy to use.

1.4 SUPPORT OF SIMULTANEOUS INCOMING/OUTGOING CALLS

SipServer open one UDP socket to listen to all SIP INVITE from all clients, and it maintains a list of clients in memory, once a session is established, it put the client id into the list of clients, the multiple RTP sessions are handled by JMF itself, whilst, when a client hands off the call, SipServer find the id of that client, and send out BYE to it, finally removes it from the list of client.