P2P Packet Voice Conferencing System
(LCT1a-10)

Students:
WONG Chung-Yin
WONG Chung

Supervisors:
Prof LEA Chin-Tau
Prof Albert WONG Kai-Sun

Introduction

Our final year project is to complete the gossip-based delay-sensitive P2P packet voice conferencing system and to improve its compatibility for all computer users. We aim to develop the system for Windows and to implement audio encoding.

System Block Diagram

Our conferencing system is constructed by several threading components shown below:

Methodology

The original gossip-based delay-sensitive P2P packet voice conferencing system was developed under the Linux environment using the cross-platform utility library SDL. However, some data types and functions are structurally different between Linux and Windows platforms. Therefore, we first looked into the selection of the libraries.

Example of library selection (internet operation libraries):

<table>
<thead>
<tr>
<th>Test</th>
<th>SDL</th>
<th>Sockets</th>
<th>Functionality</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>write</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>recv</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>send</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

In our early stage of the program migration, we were using script files which are composed by a series of “1” and “0” instead of sound recording or audio files. It is based on the experiment factors listed below.

<table>
<thead>
<tr>
<th>Task</th>
<th>Sound recording</th>
<th>Audio file</th>
<th>Script file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Modification</td>
<td>Easy</td>
<td>Hard</td>
<td>Easy</td>
</tr>
<tr>
<td>Measurement</td>
<td>Hard</td>
<td>Hard</td>
<td>Easy</td>
</tr>
</tbody>
</table>

The codes used in the digital topology is G.711, which is an ITU (International Telecommunication Union) standard. It requires 30 packets per second in order to provide good voice quality and low latency. Therefore, we used a sleep function to control the sampling interval of packet processing to be exactly 30ms.

Testing

The execution process of the conferencing system is as follows.

1. Connect all the peers to a hub
2. Connect a peer to the hub
3. A peer sends information packets

In order to have all the peers exchanging the information packets in the same cycle, the system time difference should be within 20ms. Therefore, system time synchronization of the computers under test is needed.

Results

The simulation began with 3 peers and the results were all correct. Then we added one more peer to the simulation, the number of packet received appeared to be larger than expected. By using debugging message functions, we found that the sleep function took approximately 30ms break other than the 20ms required by G.711. Therefore, we are working on solving this time function problem.

Conclusion & Future Extension

We have successfully implemented the program from Linux to Windows environment. We are working on fixing the program defects. This system can further be developed in video sharing, online games, or even VoIP broadcasting.