

CS 422

P.S.O.

October 30th, 2003



Lab 5: Building Internet Phone

- Implement a telephone (an instrument for reproducing sounds at a distance) using Internet as transmission media
- Tools:
 - Visual Studio .NET
 - Windows Sockets
 - DirectX/DirectSound
- For Windows and .NET, use the PC ITAP machines

Architecture: Main Components

- **Telephone Program**

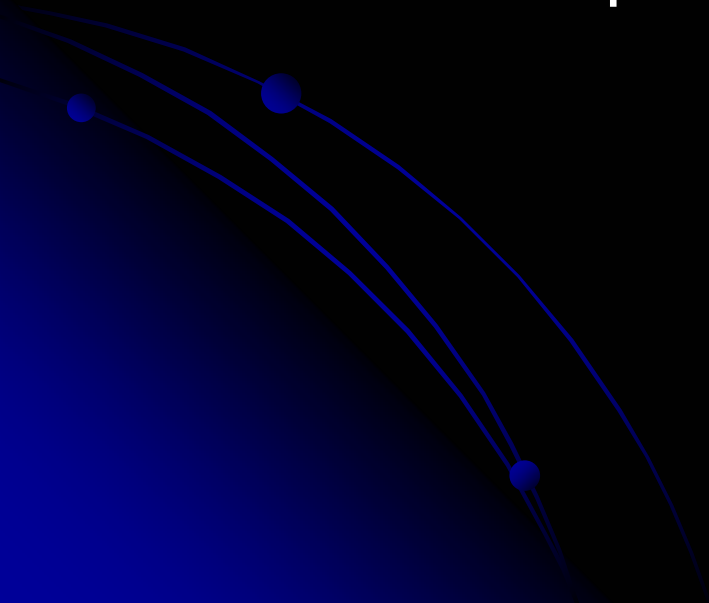
- A phone client that allows sending and receiving the calls
- Sub components:
 - Directory Client
 - Ringer Server
 - Ringer Client
 - Voice Capture Client
 - Voice Player Server

- **Directory Server**

- Maintains information list (Name/IP) about current users of the Telephone Program

Telephone Program Subsystems

- **A Directory Client**
 - Connects to **Directory Server** to add (register) the user and IP address.
 - Connects to **Directory Server** to query the list of current phone users.



Telephone Program Subsystems: Communication

- A **Ringer Server**

- Waits for incoming communications
- Notifies the user if there is an incoming communication.
- Keeps the connection open until the telephone hangs up.

- **Ringer Client,**

- Contacts the **Ringer-Server** in the other end when the dial button is pressed for placing a call.

Telephone Program Subsystems: Audio Transmission

- **Voice Capture Client**

- Captures voice from the microphone and sends voice stream to the **Voice Player Server** in the other end.

- **Voice Player Server**

- Receive the voice stream sent by the **Voice Capture Client** and plays it.
- 

Step 1: Example Code

- Internet-Phone.zip contains three examples:
- UDPTimeServer is a UDP time server using UDP sockets in Windows.
- UDPTimeClient: is a UDP time client using UDP sockets in Windows.
- Capture-Play is a program that shows you how to capture and play sounds in Windows using DirectX/DirectSound
- All the files linked off Lab 5 handout website
- All the files are .NET workspaces

Step 1: Example Code Testing

- Build and run UDPTIMEserver
- Build and UDPTIMEclient
- Run 'UDPTIMEclient localhost' in Windows command shell
 - Start → Run... → "cmd"
 - "cd UDPTIMEclient/Debug"
 - "UDPTIMEclient localhost"

Step 2: Basic GUI

- Using wizard of .NET, create a basic GUI application for Internet Telephone
 - New → Project → MFC-Application
 - Application Name: 'Internet Phone'
 - Location: the directory where you unzipped the files, i.e. where all the example projects are.
 - Application type: Dialog-Based
 - When done, click Finish

Step 2: GUI Components

- A list of all the users that are currently in the directory server.
 - Program will get this list from the directory server.
- A button to dial, answer, and hang up.
 - If there is no call in progress, the button will show "Dial".
 - If there is a call in progress, the button will show: "Hang-up".
 - If there is an incoming call, it will display "Answer"
- "Profile" button:
 - Opens another dialog box to update the name of the user and the address of the Directory Server.

Step 3: Directory Server

- Directory Server: another GUI dialog-based application
 - Has list of the users
 - Has “Shutdown” button
- The server keeps in memory the list of users
 - Each user: a pair **<name, IP-address>**
 - Users added by the Directory Clients in Telephone Program
- Directory server only deals with user list, users will communicate between each other directly

Directory Server/Client protocol

- Uses TCP Sockets
- **Adding user**
 - Happens when Telephone Program starts
 - Server waits for connection and the request to add in the form:

```
ADD<crLf>  
<Name><crLf>  
<crLf>
```

- IP address is taken from socket information returned by accept() call
- Pair <IP, Name> added to list of users

Directory Server/Client protocol

- Deleting user

- Happens when Telephone Program exits
- Server waits for the request to delete in the form:

```
DELETE<crLf>  
<crLf>
```

- IP address is taken from socket information returned by accept() call
- Corresponding pair <IP, Name> is deleted from the list of users

Directory Server Responses

- After ADD/DELETE, server generates one of the responses

- OK-response

```
OK<crLf>  
<crLf>
```

- Error-response

```
ERROR<crLf>  
<Error message><crLf>  
<crLf>
```

Directory Server Query

- **Phonebook query**
- Server expects a string:

```
GET<crLf>  
<crLf>
```

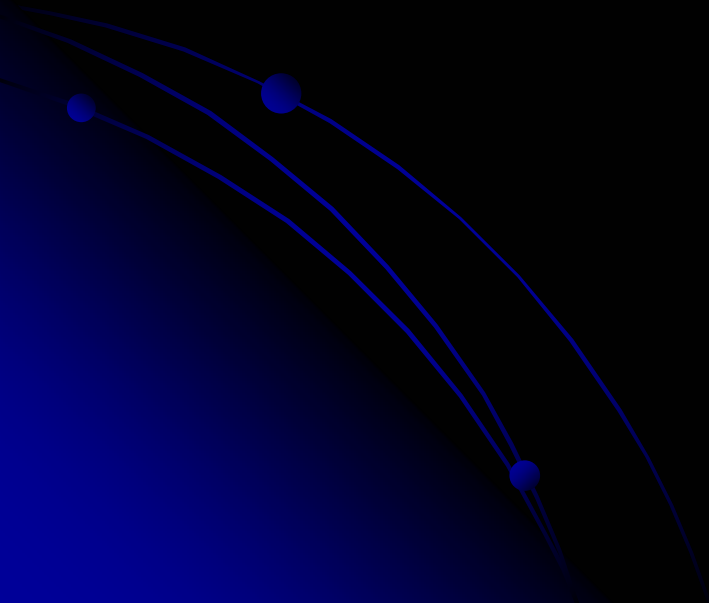
- Response is in format:

```
User1<crLf>  
ip address 1<crLf>  
User2<crLf>  
ip address 2<crLf>  
...  
Usern<crLf>  
ip address n<crLf>  
<crLf>
```

- (Or the error response in case of an error)

Step 3: Summary

- Implement Directory Server
- Implement Directory Client part (registering user and querying the phonebook) of the Telephone Program



Project Schedule

- Step 4: Ringer Server
 - Step 5: Voice Capture Client and the Voice Player Server
 - Step 6: Other things to do
 - Tune in next week
-
- Steps 1-3 are due **November 5th, at 11:59pm**
 - Zip all your files and FTP them to lab machine
 - Turnin for stage command:
 - `turnin -c cs422 -p lab5-1 Internet-Phone.zip`