Manan Buddhadev (mcb5345@rit.edu)

Hema Bahirwani (hgb1348@rit.edu)

CSCI 651: Term Project – Voice over IP (VoIP)

Introduction:

Voice over IP is the transmission of data over the network. There has been a substantial increase in the use of VoIP application in the past few years. Some of the VoIP applications are Viber, MagicApp, Skype and Google Hangouts. VoIP converts analog signals to digital signals, sends it across the internet to the receiver, and converts the digital signal back to analog. It requires good internet connection, so as to avoid lag or any anomaly.

Motivation:

SPIRIT VoIP Application is built to make internet voice calls. It encompasses the functionalities of making a call, disconnecting a call, and apprises the user if the number that it was trying to call is reachable or not. The application is built in Java using User Datagram Protocol. The user interface is built using Java swing. The Application gets its name from the University's mascot, which was a Bengal tiger cub named Spirit.

Solution Design:

VoIP converts audio received from to the microphone to bytes and then sends it over the network to the host it is communication with. At the receiver end the bytes received are re-converted into audio and played. VoIP uses Session Initiation Protocol to establish and end a connection where the audio sending/receiving is done using Real Time Transmission Protocol.

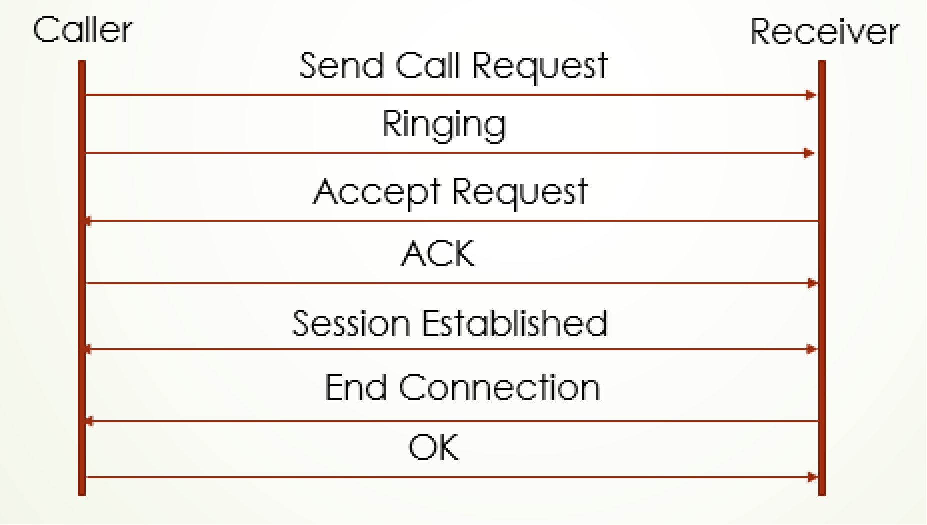

Figure 1: Session Initiation Protocol Working

In our design we have implemented SIP which send a request to the server(Receiver) and if the receiver wants to communicate, he/she will accept the call by hitting the accept button, else wait for time out or hit end.

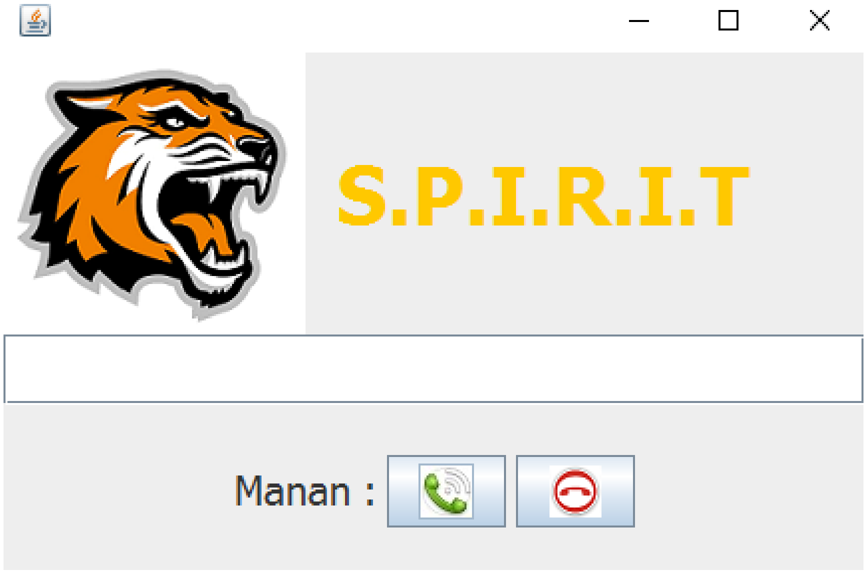

Figure 2: The User Interface: Caller

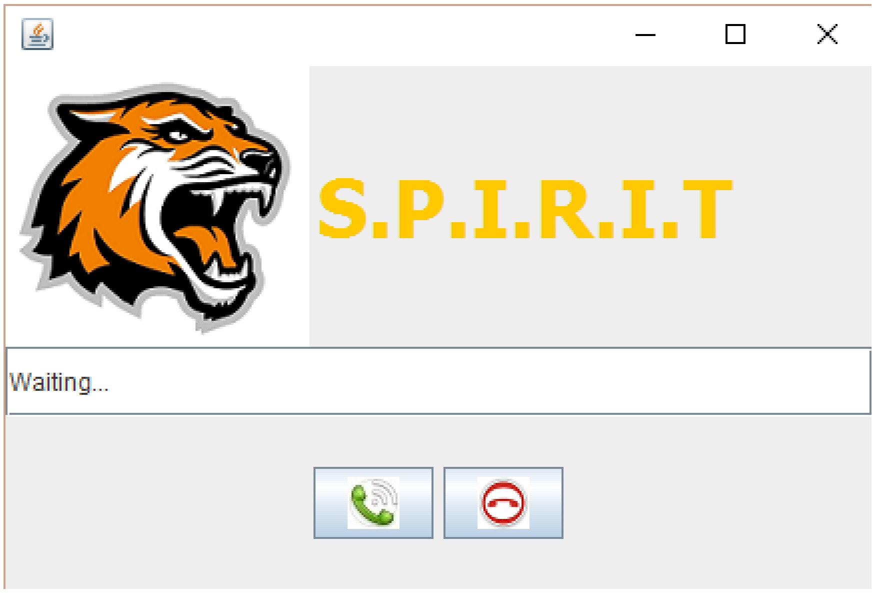

Figure 3: The User Interface: Receiver

Once the connection is established two threads to send and receive audio are started on both hosts. The voice is recorded using Java's inbuilt sound library and re-converted using the same. The audio sending thread on the host end converts the sound into bytes and send it to the other host and the audio receiving thread on the host converts the bytes received into sound to be played. After, the communication is done, either of the two parties end their connection, thus completing the voice over IP implementation.

Class Diagram:

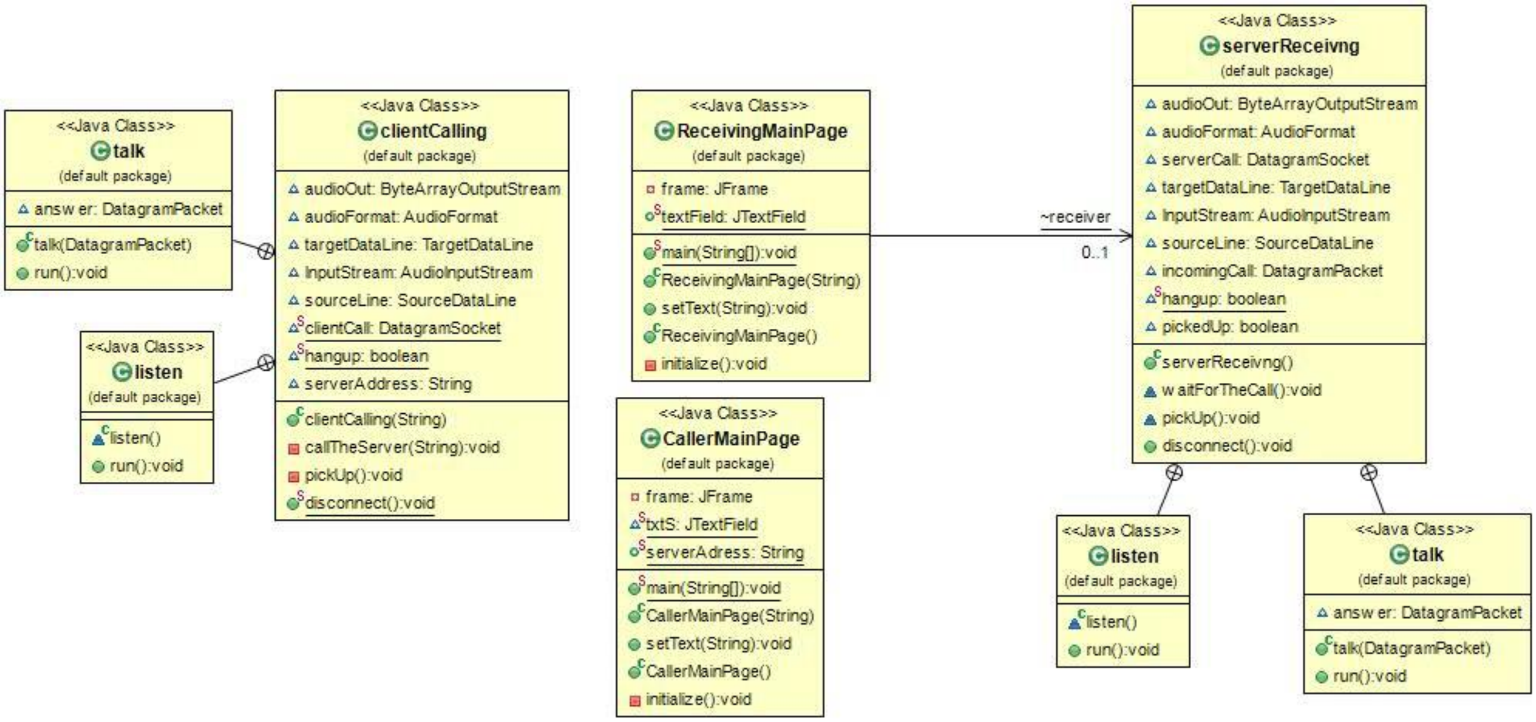

Figure 4: Class Diagram

Class Descriptions:

1. **clientCalling:**
   This class is to send the request to talk to the other host

2. **CallerMainPage:**
   This class the User Interface of the Caller

3. **serverReceiving:**
   This class is accept/reject the request received from the caller

4. **ReceivingMainPage:**
   This is the UI class for the Receiver

5. **Threads:**
   a. **Talk:** To convert audio to bytes and send over the network
   b. **Listen:** To convert received bytes to audio and play it