As broadband internet access begins to pervade American households, traditional phone lines are being replaced with a new technology called VoIP which is short for Voice over Internet Protocol. While some VoIP implementations require users to place calls through their computer (such as Skype), others allow the user to place calls using phones similar in function to traditional analog phones. Although time constraints prevent us from developing a fully functional VoIP phone system, we would like to develop a similar system for our final project. Our system will still be able to connect to the internet and transmit voice data, but it will not adhere to the VoIP standard and thus will be unable to communicate with certified VoIP systems.

VoIP is not itself a protocol as are IP or TCP. VoIP refers to a collection of protocols, all of which are designed to transmit voice over IP-based networks. These VoIP protocols are at the top of a stack of protocols used to implement IP-based networks which are routinely described through the 7-level OSI Model. Level 1 of this model consists of the physical wires used to connect network nodes, while level 7 consists of application protocols such as those that fall under VoIP. To simplify our development process, we plan to use a wireless network adapter to handle levels 1 and 2 of the OSI model (the physical and data link layers, respectively). In order to connect to the internet, however, we plan to implement handling of the IPv4 protocol (level 3) so that our device may send and receive data over the internet.

The general block diagram of our system is shown in Figure 1.

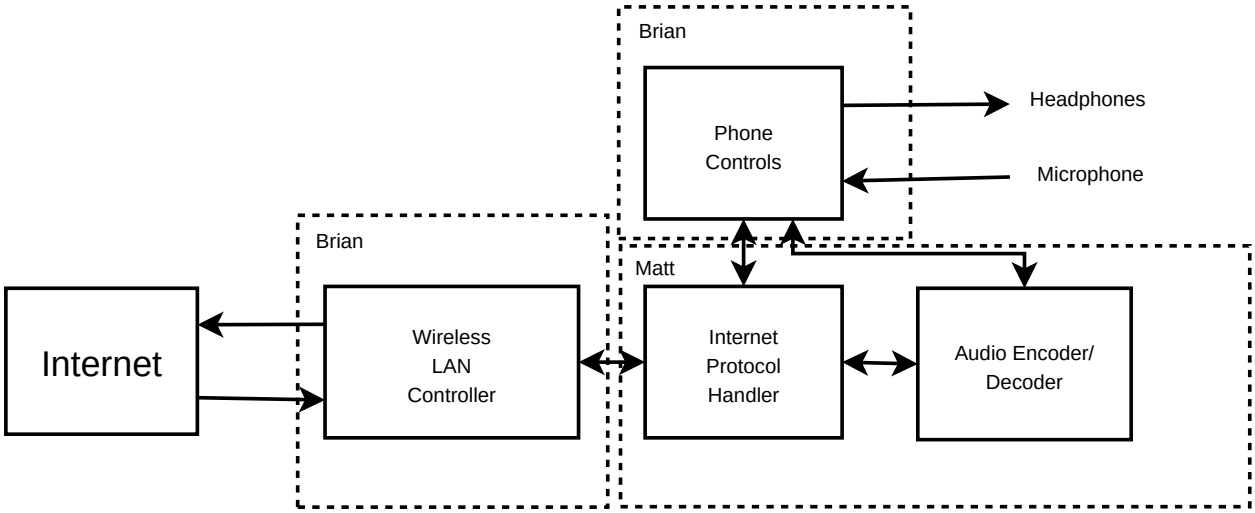

Figure 1: Project Block Diagram

The inputs to our system will consist of audio input from a microphone via the AC97 chip, buttons for user input, and data packets from the network. We will use an audio encoder to compress the audio data before transmitting it over the network, and a complementary audio decoder to decompress the audio data once it is received from another phone.

The outputs of our system will consist of audio output to the headphones via the AC97 chip, lights to indicate connection status, and data output from the wireless LAN adapter.

Testing of the wireless controller will be done using a network analyzer to listen to and validate the data packets being sent to and from the controller. Since we will be using a third party wireless network adapter and most of the work will be put into writing the interface and drivers for the controller, simulation will play a key role in the testing of the wireless network adapter. All of the interface drivers will be tested by running simulations and comparing the data to the expected results based on network specifications before being tested in the field.

Wireless Network Adapter

This module describes both the physical adapter and the drivers used to control it. In our original design, we planned to use wired ethernet, and an open source ethernet controller written in verilog. Instead, we have decided to pursue a wireless design, to facilitate ease of use. We hope to receive funding in order to purchase two wireless network adapters, for use in our project. We will then acquire data specifications for the devices, and write drivers which will interface with the controllers. After researching wireless protocols, we have found that using a third party adapter and writing our own drivers will prove to be a more than adequate digital design challenge. There are many layers of protocols that exist on top of the networking controller, and this module will sort through them to make sure the data is sent and received properly over our two wireless network controllers.

Brian will be responsible for implementing the wireless network adapter interface.

Phone Controls

This module will contain all of the actual user interfaces. There will be buttons to both call the other user, and to accept an incoming call. It will have a microphone as an input, and headphones as an output. The user will also be able to view the call status and network activity on the labkits LEDs.

Brian will be responsible for implementing the phone controls module.

Audio Encoder/Decoder

Initially, our Audio Encoder/Decoder will be a simple sampling of PCM data produced by the labkit's AC97 chip. The sampling rate will be chosen to be compatible with the limit we're able to transmit and receive data to and from the wireless network controller. If time permits, we hope to implement either the G.711 codec or the more complex G.729 codec which are both commonly used for VoIP.

We plan to test our encoder/decoder by feeding audio data from a microphone to the encoder, passing the encoder's data through the decoder, and then listening to the result through a speaker or headphones. This ensures that any data we send to another phone will be able to produce an intelligible voice signal.

IP Handler

The IP Handler will convert audio information from the audio codec module into packets to be sent over the network as well as convert packets received from the internet into compressed audio data to be sent to the audio codec. In addition, it will handle signals to and from the phone controls such as establishing a connection, disconnecting, and indicating connection status.

We may test the IP Handler by storing its output to memory on the 6.111 Labkit and then sending the data back through the handler as if it were receiving the data from another phone. This circumvents the wireless network controller, so that the IP handler may be tested independently. If the received signals produced the desired output in the phone and the expected audio signals, we will know that the module is working.

Because the IP protocol is a standard and widely-used protocol, we may consider analyzing the data send from the labkit on a computer with a wireless connection once the wireless network adapter is functional. This may be done using a packet sniffer or network analyzer on the computer with the wireless connection.

Matt will be responsible for implementing the IP handler module.

While this project may seem complex overall, we feel that the complexities are manageable, and that we will be able to produce a functional system in the time provided.