Project Number: DRB-502

Many-to-Many Digital Wireless Music Distribution System

A Major Qualifying Project Report
submitted to the Faculty
of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the
Degree of Bachelor of Science
by

Paul Kastner

J. Michael McHugh

Stephen O'Dea

Andrew Papia

April 26, 2006

Professor D. Richard Brown III, Advisor

Contents

List of Figures

List of Tables

Abstract

The goal of this project was to design and implement a many-to-many digital wireless audio distribution system. While other wireless audio technologies exist, none offer the unique combination of many-to-many scalability, lossless digital transmission, source control, and universal connectivity to home audio equipment. As our primary deliverable, we developed a two-by-two proof of concept of this system on x86-based single board computers with Wi-Fi capability. To enable the functionality of our system, we developed software enabling the server units to distribute audio data to client units over an ad hoc network using multicast. We also implemented an LCD-based user interface, and designed a wireless means for the transmission of infrared remote control commands back to the audio source. Testing was performed to verify the range and scalability of the wireless link.

Executive Summary

The ability to listen to music from a single source in multiple locations provides a useful and enjoyable enhancement for many home audio systems. However, distributing music throughout the home has proven to be logistically difficult. While home owners can extend the listening range of an audio source by purchasing a multi-room amplifier, this requires the installation of additional wiring throughout the house by a professional electrician. For many consumers who do not own their homes or who cannot afford such an installation, this solution would be infeasible. The lack of a practical, affordable technology for distributing music throughout the home has resulted in the majority of consumers being confined to listening to their music in the same room as their audio source.

While there has been a focus on the development of portable audio products for several years, companies have just recently begun to create products which allow consumers access to music throughout a home. This recent boom in home-wide music distribution systems has yielded a range of products, each of which address only a particular subset of consumer needs.

This project developed a product that incorporated a set of features for which consumer demand is visible, but is different from any currently available product. These features include universal connectivity to audio sources and sinks, digital wireless transmission, many-to-many scalability, an intuitive user interface, and source control from any receiver unit.

This feature set allows consumers to use their existing audio equipment and music libraries while enjoying their music at any desired location in their home. In contrast to traditional systems, this functionality is achieved without the need to install costly or inconvenient communication infrastructure in their home. Furthermore, the feature set allows multiple users to access the system simultaneously from various locations throughout a household, ensuring that the need for a truly home-wide music distribution system is met. Figure 1 shows an example of how such a system could be implemented in a house. Transmitter modules (denoted by TX) are connected to music sources and receiver modules (denoted by RX) are connected to speakers.

To demonstrate the feasibility of such a feature set, we developed and implemented a proof-of-concept system. This system shows how available technology can be utilized to enable a wide range of functionality. This prototype system also allowed us to conduct performance testing of the design. Data from these tests illustrated the relationship between distance and packet loss at two different transmission frequencies.

In order to allow users to not only listen to but control audio sources remotely, an IR repeater was developed. This repeater allowed users to control audio sources using IR remote-controls supplied with audio products from any location in a home where the system is in use. The IR repeater was designed to ensure interoperability with all commonly available IR remote-controls.

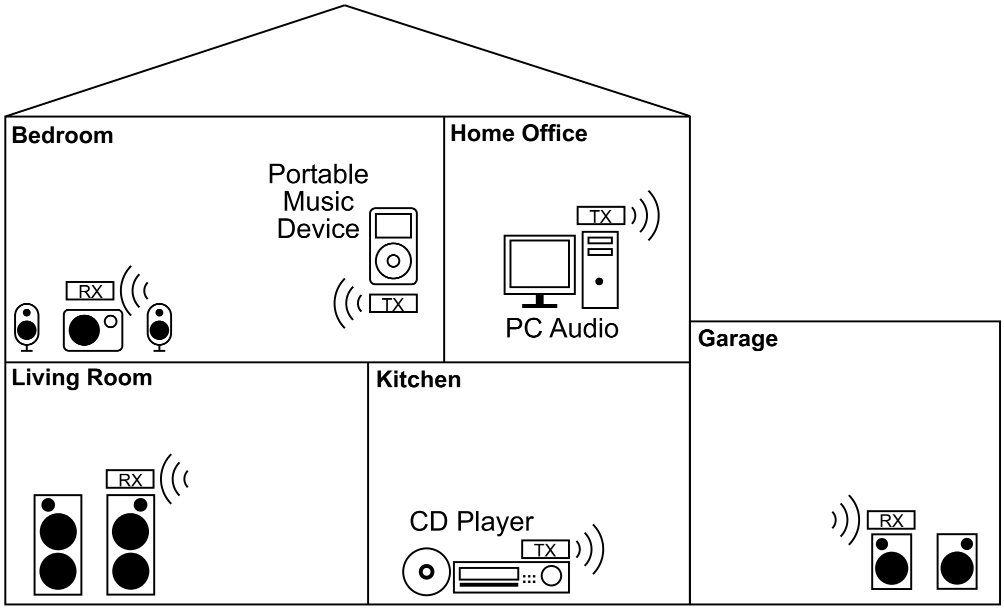

Figure 1: A example implementation of a many-to-many music distribution system in a house

The proof-of-concept system utilized a wireless link to transport music between audio sources and listening systems. Due to congestion of the RF spectrum, the throughput of this link is restricted. In order to effectively utilize the available throughput, the transmission of redundant or unnecessary data was minimized. This led to a system design which simultaneously allowed uncompromised audio quality and efficient wireless spectrum usage.

One of the significant factors in determining the performance of the system was the quality of the wireless link. In order to characterize the performance of the system, thorough testing of the wireless link was performed under varying conditions. This testing revealed a range of factors which yielded ideal performance of the system as well as conditions that can potentially cause undesirable operation. This testing has led to the creation of a number of recommendations which can be used to improve the performance of the system. Ideas for future development of this system have been broken into three categories: general product improvements, further testing and dealing with packet loss.

Consumer trends clearly indicate the need for the features implemented in this system. The availability of off-the-shelf components to implement the system ensures that the complete system can be supplied to consumers at a reasonable price. Furthermore, the recent boom in wireless home-wide audio solutions has created the consumer awareness necessary to facilitate rapid market adoption and commercial success.

Acknowledgments

Our group has been fortunate to have received guidance and assistance from a number of individuals during our work on this project. First, we would like to thank our sponsors at Bose, Steve Christensen and Jeff Faneuff, for presenting us with this opportunity, as well as providing interim feedback. We would like to thank Christian Trott at Bose for helping us with our wireless research

We would also like to thank a number of members of the WPI community for their efforts in providing us direction and focus on the goals of this project. First, we would like to thank Professor Rick Brown, for advising our project and offering us guidance throughout the design and implementation process. We would like to thank Professors Stephen Bitar and Sergei Makarov for providing their time to offer technical support in the analog hardware design and antenna selection processes, respectively. We would also like to thank Tom Angelotti in the ECE shop not only for helping us find circuit components, but also for assisting us in fitting the LCDs and PCBs into our enclosure. We would also like to thank Network Operations at WPI, specifically Joseph Krzeszewski and Sean O'Connor for providing us with areas free of other wireless networks (for testing our system). We would also like to thank the work-studies at Washburn Shops for their assistance in modifying our enclosure.

Finally, we would like to thank Richard M. Stallman for guiding us in the creation of not only technically but also ethically good software.

Chapter 1

Introduction

Home audio systems can be enhanced and made more versatile by incorporating the ability to listen to music from a single source in multiple locations. However, distributing music throughout the home has proven to be logistically difficult. Home owners could extend the listening range of an audio source by purchasing a multi-room amplifier and having additional wiring installed throughout the house by a professional electrician. However, for many consumers who do not own their homes or who cannot afford such an upgrade, this solution would be impractical. The alternative of having interconnecting wires laid out visibly throughout the home has also been unappealing to most consumers for reasons of safety and aesthetics. Therefore, the lack of an available, affordable technology that would overcome the need for wired audio device interconnection greatly restricted the home audio possibilities for most people.

Recently, wireless systems have become available that offer home wide music distribution that is in many cases more practical and more elegant than the wired methods previously mentioned. For example, companies such as Roku, SMC, Philips, D-link, and Linksys offer media adapter products that allow the user to access digital music stored on a computer from a home entertainment system using an existing wireless network. Computer-stereo link systems such as these are limited in that they can only play back music that is stored on the host computer. These systems also deprive the host computer of resources such as hard disk space for the storage of music as well as processor power and RAM which are needed to continuously run the server application. Additionally, they require that a wireless network be set up in the home which demands a reasonable amount of effort from the consumer if the network is not already in place.

Alternatively, analog-stereo link systems allow the user to connect their existing home audio equipment via a wireless link. Some of these products, for example the RF Link AVS-5811, allow transmission of multiple audio streams on separate RF channels. They may also incorporate an infrared (IR) repeater to provide control of the audio source. Many of these products suffer from the losses associated with analog transmission methods which are inherently susceptible to noise and interference from other wireless devices. This level of signal degradation is often noticeable with current high-quality audio playback mediums. Therefore, any device which handles audio between the playback system and the listener should maintain the integrity of the original signal.

While there is a large range of functionality available in current wireless music distribution systems, consumers must often make compromises when selecting a product to suit their specific needs. For example, the majority of products available to consumers wishing to stream music digitally only allow streaming from a computer's music library to a single receiver. One may alternatively choose an analog system to connect multiple audio sources but the audio quality will be degraded. Also, the maximum number of usable sources is limited to the number of available channels, which may be reduced if nearby systems are operating on the same set of frequencies.

This project involved the design of a digital wireless music distribution system that addresses the shortcomings of similar consumer products that are currently available. Unlike the computer-stereo link systems discussed, any standard audio source or audio sink with stereo line level connections can be connected to a transmitter or receiver in this system. In addition, the user can control the audio sources from the receivers using its standard infrared remote control. Also, unlike the analog-stereo link systems, this system is able to transmit lossless digital audio from many sources to many receivers simultaneously. Any available source can be selected from each receiver making this truly a many-to-many system.

For this project, a proof-of-concept system with two transmitters and two receivers was constructed to demonstrate the feasibility and functionality of the system. Given the 54 Mbit/s theoretical throughput of the 802.11 a/g wireless links used in the proof-of-concept system, our analysis suggests that the system could support up to 38 separate streams of uncompressed audio. This report documents the design and functionality of our product, offers recommendations for future development, and provides suggestions for making it production feasible.

Chapter 2

Background

The intent of this chapter is to provide the reader with information pertinent to understanding our product design. This background chapter is divided into four major sections.

The first section of our background examines wireless home audio devices currently on the consumer market. We did this in order to understand current applications of wireless technology in home audio applications and also to scope a product with unique features.

To establish our system's throughput requirements, the second section of the background explains the digital audio format used in this system. It also describes the physical connections common to many home audio devices.

The third section of this chapter deals with radio frequency (RF) wireless transmission. It examines the restrictions on wireless transmission put forth by the Federal Communications Commission (FCC), which are vital to understand in order to assure our product's compliance with these regulatory codes. It also describes the existing wireless protocol standards for IEEE 802.11, which were used for the implementation of this system.

The final section of this chapter examines infrared (IR) remote control standards, including wavelength and modulation frequency, since the system provides source control through infrared commands. It describes how infrared light is generated and modulated. The information in this section is provided for understanding the design of our IR repeater hardware.

2.1 Prior Art

There are many wireless audio products on the consumer market today but the vast majority fall into two general categories. The first are analog wireless audio electronics that transmit analog audio signals using RF modulation. The second are digital audio products that use Wi-Fi or Bluetooth for wireless connectivity. The following sections will describe the general feature set of the products in these categories and cite specific product examples.

2.1.1 Analog Wireless Audio Electronics

There are several analog solutions for home audio distribution, of which wireless speaker systems are common. Typical systems such as the RCA WSP150 900 MHz Wireless Speakers are simply speakers with an integrated analog RF receiver. A base station with an RF transmitter is connected to the audio source. These systems provide only point-to-point wireless connections and do not allow for the use of generic audio sinks nor control of the audio source from the listening location.

Figure 2.1: RF-Link AVS-5811 wireless system[1]

There are more sophisticated analog systems, such as the RF-LINK AVS-5811 shown in figure 2.1, that allow for home wide audio distribution. The AVS-5811 can transmit audio and video signals on four selectable subbands in the 5.8 GHz band. It provides connectivity to generic audio and video equipment via RCA style analog inputs and outputs. Source control is provided through an IR remote extender which allows the user to use the source's remote control at the location of the receiver. The AVS-5811 is limited, however, to a maximum of four audio/video streams on the four RF subbands. The AVS-5811 also suffers from signal distortion from RF interference which is a problem common to all analog wireless systems[1].

2.1.2 Digital Wireless Audio Products

Most of the wireless audio products emerging on the market today are digital systems. Digital wireless systems have the advantage over analog systems of immunity to distortion due to RF interference. There are also relatively new digital wireless standards such as Wi-Fi (IEEE 802.11) and Bluetooth that are gaining popularity and making digital wireless systems cheaper to produce and easier to develop. Digital wireless solutions are also well suited for applications where the data stream is already digitized, such as when interfacing with an MP3 music library.

Figure 2.2: Apple AirPort Express with AirTunes[2]

Recently, a number of companies have released digital wireless audio products for accessing a digital music library on a computer from the home entertainment center. Some of these products are the SMC EZ-Stream Wireless Audio Adapter and the Apple Airport Express (figure 2.2)[3][4]. Most of these systems only offer a point-to-point connection between a speaker system and a PC with music library software and a Wi-Fi card. Inherently, these systems are limited in that they do not support connection to generic audio sources and require an existing 802.11 wireless network.

Figure 2.3: Bose Link transmitter and receiver units[5]

Another interesting digital wireless audio product is the Bose Link AL8 Wireless Audio Link (figure 2.3). This audio distribution system is designed to enable digital wireless audio transmission from a Bose Link transmitter module to a Bose Link receiver module up to 80 feet away. Up to eight separate receiver modules can communicate with a single transmitter module at one time. Rather than using an existing wireless standard such as Wi-Fi or Bluetooth, Bose has developed its own proprietary protocol that minimizes interference from other 2.4 GHz sources.

While Bose has taken an innovative approach to the wireless audio distribution system, there are still several shortcomings. First, it is a closed system that has only limited support for the use of generic audio sources and sinks. The Bose Link system must use a Bose Lifestyle system as the audio source. A generic source can only be used by connecting it to the Lifestyle's single auxiliary input. Also, source control is provided through the use of an RF remote for the Bose Lifestyle Home Theater System which must be purchased separately[5]. Table 2.1 provides a means for a quick comparison between some of the mentioned products.

Signal Paths

A very basic block diagram of the wireless music system is presented in figure 2.4. While this figure does not illustrate the many possible signal paths between many audio sources and many audio sinks, it does show the basic signal flow from a source to a sink. There are two particularly important signal paths to consider here. The first is that of the analog audio signal which comes from a music source or goes to an audio sink, as indicated by the solid lines. The other signal path is the wireless propagation of the audio from the transmitter unit to the receiver unit of the system. The dotted line arrow indicates this signal path.

The next two sections of this chapter provide background information regarding the propagation of the audio information through these signal paths. The first of the two sections describes the

Table 2.1: Comparison of proprietary wireless systems

	Bose Link AL8	Apple Airport Express	RF-Link SV-5811	SMC EZ-Stream
Maximum range	80’	150’ at 11 Mbps, 50’ at 54 Mbps	300’ line of sight	150’
Price	$399 for transmitter and receiver $149 for additional receiver	$129 for receiver unit	$249.95 for transmitter and receiver	$110 for receiver unit
Operating Frequency	2.4 GHz	2.4 GHz	5.8 GHz	2.4 GHz
Transmission Method	Proprietary Digital	802.11g Digital	Analog	802.11b Digital

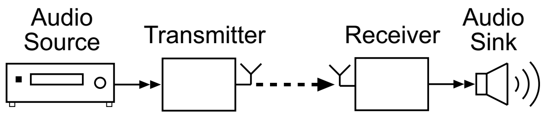

Figure 2.4: Audio signal path from source to sink

characteristics of the digitized audio signal. The following explains the wireless frequency spectrum and the particular wireless standard used for wireless communication.

2.2 Audio

Most common audio sources and sinks have analog audio outputs and inputs respectively. Normally, the analog connectors on these devices are stereo RCA jacks which allow connectivity of line level audio signals. Line level audio is a low power signal that is designed for high impedance loads. This signal is intended to transfer the audio from one device or circuit to another. Additionally some devices, such as portable CD and MP3 players, provide an $\frac{1}{8}$” headphone output. The signal from this connection, unlike line level audio, is intended to deliver power to low impedance loads. The system must be able to accept both types of signals and to produce line level signals to ensure its compatibility with generic sources and sinks.

To preserve audio quality, our product will handle audio signals digitally. To obtain digital audio from a generic analog source, the analog audio signal must be sampled and converted to a digital representation. An analog to digital converter (ADC), like the one shown in figure 2.5, performs this operation. The sampling frequency and resolution of the ADC, which can be varied as required for a particular application, determine the bitrate of the digital audio stream. In this application, audio is captured at CD-quality. This implies a sampling rate of 44.1 KHz and a resolution of 16 bits. Two such channels of this digitized audio, comprising a stereo stream, correspond to a total data rate of

$$2(44100\frac{samples}{s} \times 16\frac{bits}{sample}) = 1411200\frac{bits}{s}$$

In other words, the digital transfer of each stereo audio stream requires 1.41 Mbps (Megabits per second) of throughput over the transmission medium.

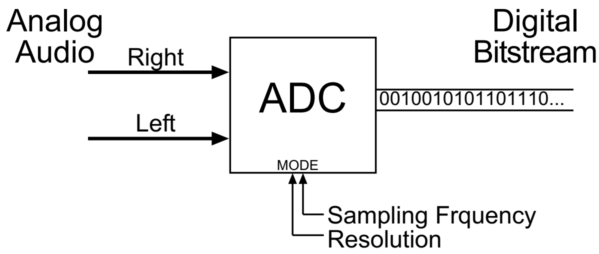

Figure 2.5: Analog to digital conversion

2.3 Wireless Regulations

In order to select an appropriate radio frequency for wireless transmission, it was crucial to understand the government regulations that apply to RF radiating devices. Since this system was developed in the United States and primarily for use in the United States, only the US regulations were considered for this project. The Federal Communications Commission (FCC) regulates all RF transmissions in the United States. Many bands of the frequency spectrum require a special license from the FCC in order to transmit above a certain power level. The FCC also defines parts of the spectrum that can be used without a license provided the device does not exceed the specified field strength and any additional provisions for the band or application are met. The requirements for unlicensed transmissions are defined in the FCC Code of Federal Regulations Title 45, Part 15[6].

2.3.1 Intentional Radiator Field Strength Limits

Subpart C of the Part 15 regulations defines the field strength limits for intentional radiating devices such as a wireless transmitter. Figure **??** shows the maximum field strength (in $\mu V/m$) versus frequency for intentional radiators as defined by these regulations[6]. Note that the restricted bands in which no transmissions are allowed are not shown. The highest field strengths below 10 GHz are around 900 MHz, 2.4 GHz, and 5.8 GHz and are known as the Industrial, Scientific, and Medical (ISM) bands. The ISM bands are the only frequency ranges available with enough bandwidth and permitted field strength for the high throughput wireless connection required for this project.

2.3.2 Unlicensed use in Industrial, Scientific, and Medical (ISM) Bands

Unlicensed use in the ISM bands are subject to the following provisions defined in FCC Rule Part 15.247. These special provisions apply to frequency hopping and digitally modulated intentional radiators. Frequency-hopping systems operating in the 2400-2483.5 MHz band must employ

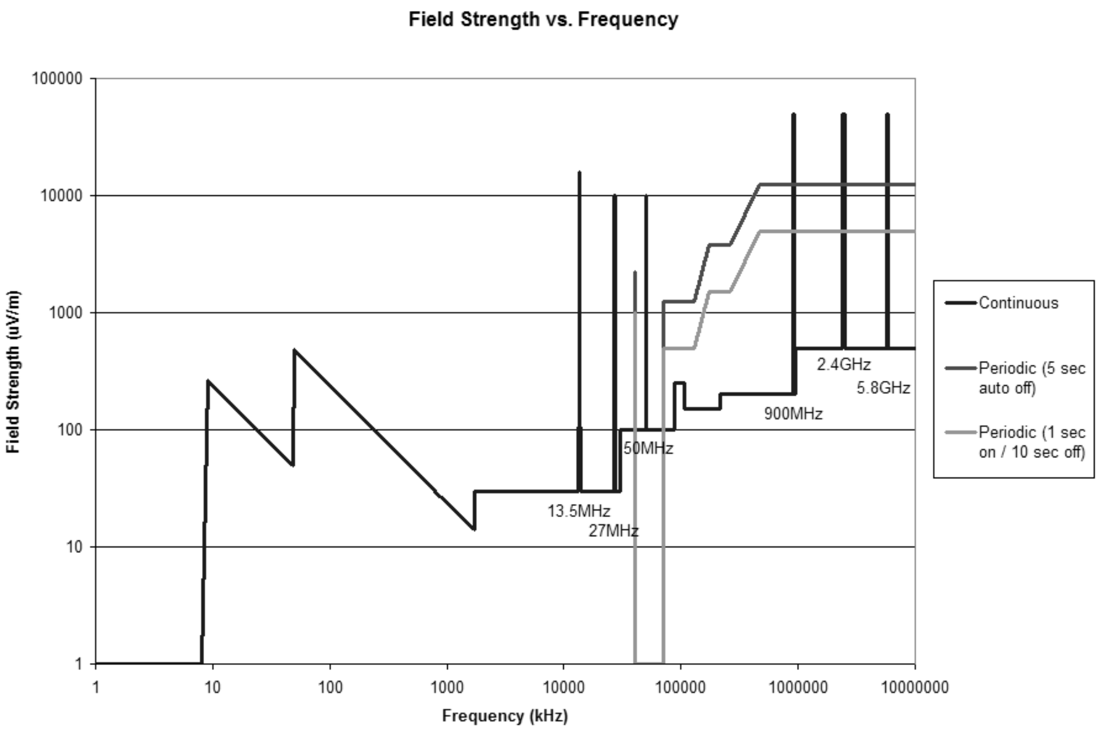

Figure 2.6: Field strength in $\mu V/m$

at least 75 non-overlapping hopping channels while operating in the 902-928 MHz band requires at least 50 hopping channels. All frequency hopping and digital modulation systems in the 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz bands can have a maximum peak conducted output power of 1 watt. These special provisions for digital modulation in these three bands make them especially attractive for digital transmission systems. The wireless standard used in this project, IEEE 802.11, uses digital modulation in the 2.4 GHz and 5.8 GHz ISM bands.

2.3.3 Wi-Fi: IEEE 802.11

The wireless music distribution system employs IEEE 802.11 for wireless communications. IEEE 802.11, also known as Wi-Fi, is the set of standards for wireless local area networks (LANs) developed by the IEEE LAN/MAN Standards Committee. The 802.11 standard currently has three different popular variants denoted 802.11a, b, and g. IEEE 802.11b was the first widely adopted standard for wireless networking. The 802.11b standard uses the 2.4 GHz band and has a maximum raw data rate of 11 Mbit/s. The 802.11a standard uses the 5 GHz band with a maximum raw data rate of 54 Mbit/s. IEEE 802.11g is an improvement on the popular 802.11b standard that supports up to 54 Mbit/s in the 2.4 GHz band[7].

Configurations

All the 802.11 variants support the same basic modes of operation. Wi-Fi networks can operate in either infrastructure or ad-hoc modes. Most Wi-Fi networks utilize infrastructure mode in a point-to-multipoint configuration that utilizes one or more access points. In this configuration, all the network traffic passes through the access point(s) which handle all the packet routing. This way, each client only needs to maintain a link to the nearest access point.

Ad-hoc is an alternative mode supported by 802.11 which allows clients to connect to each other in a point-to-point configuration. The largest drawback is that the each client must transmit directly to each of the other clients requiring each of the clients to be relatively close to each other. This problem can be overcome by implementing a mesh network that enables packets to hop from the initial client through its nearest clients until it reaches the distant destination client. This can dramatically increase the local area of an ad-hoc network. However, implementing a mesh network requires complex routing algorithms that add additional overhead to the ad-hoc network.

CSMA/CA

The 802.11 standard uses the carrier sense multiple access with collision avoidance (CSMA/CA) network control protocol to improve the reliability of data transmissions. Collisions occur when two devices try to transmit a message at the same time, resulting in both messages being garbled. CSMA reduces the probability of collisions occurring by using a carrier sensing scheme. CSMA/CA is important because it allows the network to scale reliably with additional nodes and not suffer greatly from internal interference.

The drawback of CSMA/CA is that the overhead required significantly reduces the maximum throughput achieved by each node. IEEE 802.11b can only realize a maximum of 5.9 Mbps (Megabits per second) over TCP and 7.1 Mbps over UDP out of the maximum theoretical 11 Mbps due to CSMA/CA overhead[8].

OFDM

IEEE 802.11a differs from 802.11b/g because it operates in the 5 GHz band and uses orthogonal frequency-division multiplexing (OFDM). OFDM is a highly complex transmission technique that broadcasts multiple signals at the same time at slightly varying frequencies called subcarriers. The result is a signal much more immune to interference and noise and capable of much higher data rates. IEEE 802.11a uses 52 subcarriers and can operate at 8 different data rates depending on signal strength and interference conditions[9]. 802.11a's use of OFDM appears to be why it is able to achieve lower packet loss than 802.11g in the performance test results found in Chapter 4.

2.3.4 Networking

Modern computer networks are based on the Open Systems Interconnection Reference Model (OSI Model) which is a layered abstraction of a computer network design. The OSI model has seven layers that define the network protocol from the application using the network to the physical networking device. These layers are shown in figure 2.7 below.

This project is concerned mostly with the four lowest levels of the network protocol. The physical layer is the actual device that transmits the analog signal, in this case, an 802.11a/b/g wireless networking card with an antenna. The data link layer handles the physical addressing of information and low level error detection and correction. The data link for this network is Ethernet. The network layer provides all the network routing and flow control and is defined by the Internet

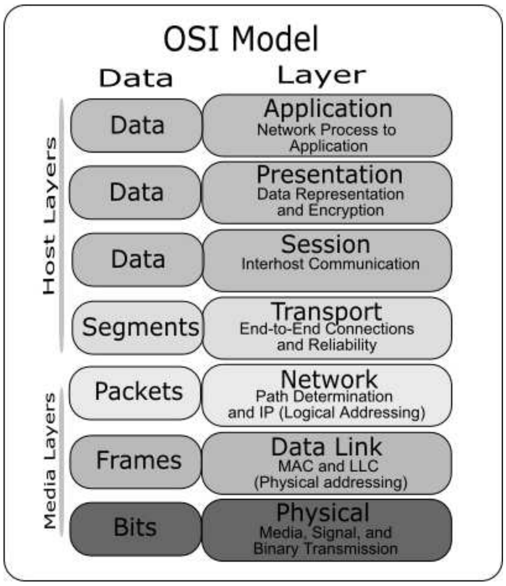

Figure 2.7: OSI model[10]

Protocol for this network. The fourth layer is the transport layer defines the protocol for reliable packet transmission[10].

The most common and widely used transport protocol for Ethernet networks is the Transmission Control Protocol (TCP). TCP works to provide reliable packet transmission by creating connections between applications over which they can exchange information with guaranteed in-order packet delivery. TCP adds a number of bits to each packet (such as a sequence number and a checksum) to ensure that the packet is not corrupted, lost, or delivered out of order. Each packet sent is then either acknowledged or the packet is resent after a timeout.

While TCP efficiently ensures that each packet is properly received, it has a number of drawbacks. First of all, it uses more bandwidth to acknowledge every packet sent and resend missing or corrupted packets. It also requires that a connection be maintained between each application. This means that if a server is streaming data to more than one client, each client must make an independent connection with the server meaning the server must send the stream to each client individually.

Additionally, TCP was not designed for use over wireless networks. Any packet loss is assumed to be from congestion over a wire so TCP scales back the transmission rate. However, wireless networks are subject to sporadic losses due to fading, shadowing, and other conditions that are not the same as congestion. Therefore, TCP has a tendency to limit the possible throughput over a wireless connection[11].

As an alternative to TCP, the User Datagram Protocol (UDP) is a common transport protocol used for sending short data packets called datagrams. UDP is designed to be simple and lightweight so it does not make use of connections or states and provides no reliability or packet ordering guarantees. As a result, UDP is often faster and more efficient for lightweight or time-sensitive applications. Since packets can go missing or arrive out of sequence, the application must be designed to function over an unreliable transport link.

A major advantage of UDP for this application is that it supports broadcasting and multicasting of packets. UDP datagrams can be broadcast to a local subnet so that all devices on the subnet will hear the message. However, in many cases only certain devices on a local network are interested in receiving the data packets. Multicast goes a step further by selectively sending datagrams to all members of a multicast group. Devices on a local network can join the multicast group by listening to a certain multicast address. In this way, packets only need to be sent once to be received by all members of the multicast group. This method is far more efficient than multiple TCP connections but at the cost of guaranteed packet delivery[12].

A number of multicast protocols have been designed to deal with UDP's unreliable nature. One popular protocol for delivering audio and video is the Real-time Transport Protocol (RTP). This protocol provides sequence numbering, time stamping, delivery monitoring, and payload-type identification of UDP packets. The protocol's main advantage is that it provides consistent order of packet delivery while still supporting multicast streaming for real-time applications. However, RTP does not retransmit packets that are lost during transmission[13].

2.4 Infrared Remote Control Basics

A simple way of providing source control for the wireless music distribution system was to make use of the infrared (IR) remote controls common to most home audio electronic equipment. These remote controls use IR light to transmit a message signal from the remote to the audio source. An infrared light emitting diode is often used to produce the IR signal. A digital control message is encoded by varying the width of high and low pulses. The electronic equipment receives the message using an infrared phototransistor and executes whatever command has been hard-coded to correspond with a given combination of ones and zeros.

While the duration of the baseband binary pulses is generally around 0.5 ms, the information is modulated on a high frequency carrier wave in order to avoid interference from ambient IR light. The frequency of this carrier wave varies from about 38 KHz to 50 KHz for most home electronic equipment. Figure 2.8 shows the carrier wave modulated in an 'on-off keyed' manner by the binary information. The incoming signal is decoded by the audio equipment through a low pass filter, which removes the carrier wave while retaining the baseband message.

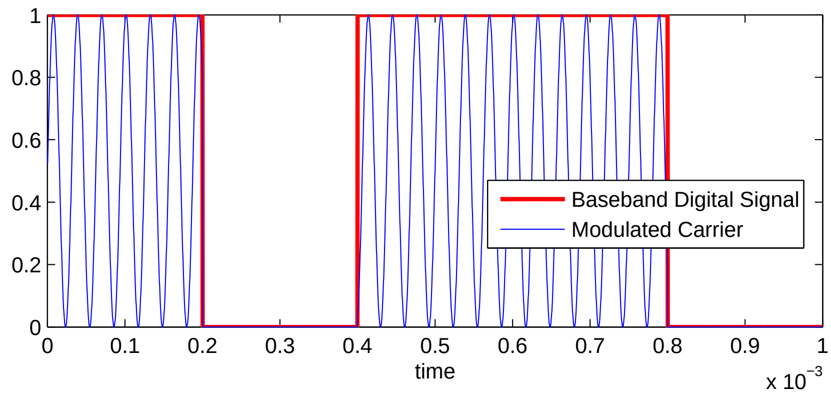

Figure 2.8: Infrared signal with carrier wave

Chapter 3

Methods and Solutions

This chapter describes the framework of our product and the design decisions we made that allowed us to meet these requirements. This chapter begins with a list of technical specifications that served as a primary guide in the research and selection of the hardware platform for the two-by-two system. Additionally, this chapter discusses the hardware selected for our infrared extender and our user interface. The software section contains an overview of our operating system choice and an explanation of the applications that were created for the transmitter and receiver units. Procedures and setup parameters for our system performance tests are given in the final section of this chapter.

3.1 Technical Specifications

One of the preliminary steps in the development of our product was the creation of a list of product technical specifications. These specifications were based on background research and the initial product requirements as described in Appendix B.

- **2.4 GHz or 5.8 GHz frequency band usage** - This requirement is based on background research regarding FCC regulations, described in §2.3. In order to allow the throughput needed to stream uncompressed audio, it is necessary to use one of the three ISM bands. We decided that 900 MHz would not be suitable for several reasons. Most notably, it had the narrowest available spectrum for transmission and it could not be used internationally (Europe reserves the 900 MHz band for GSM phones and military use). This left us with either the 2.4 GHz or 5.8 GHz bands, both of which we extensively tested for range and potential throughput.

- **Analog stereo input and output** - Almost all home audio sources offer an analog stereo (or at least single-channel analog) output. Additionally, almost all audio sinks have an analog stereo input. Therefore, in order to ensure compatibility with common home audio equipment, the product will need to have analog stereo inputs and outputs.

- **Infrared Repeater** - In order to meet the requirement of having source control from the receiver module, we decided that having an infrared (IR) repeater would be an appropriate requirement. This allows the user to control the source from the receiver using an IR remote control which is supplied with the vast majority of audio sources. This extender relies on a radio frequency connection in the 433 MHz band independent of all digital wireless audio transmissions to avoid any possible interference.

- **Intuitive User Interface** - An LCD module with buttons will provide the units with basic input and output. The user interface allows for the delivery of messages to prompt the user and assist in source selection.

- **Minimum 20 m range** - This requirement was based primarily on prior art research. While the ranges of current products vary, it is evident that the minimum requirement to be considered a home-wide wireless audio solution is at least 20 m. Implied in this requirement is also the ability to transmit through walls.

3.2 Hardware

Some time was spent selecting appropriate hardware to develop our many-to-many wireless music distribution system on. An interconnection diagram is provided below in figure 3.1. The central hardware component is an X86 single board computer (SBC), to which a wireless network adapter and an LCD module are connected. Infrared repeater circuitry for client and server machines is also depicted. This section will describe, in detail, each of the chosen hardware components as well as circuitry and enclosure designs. A bill of materials is available in Appendix C.

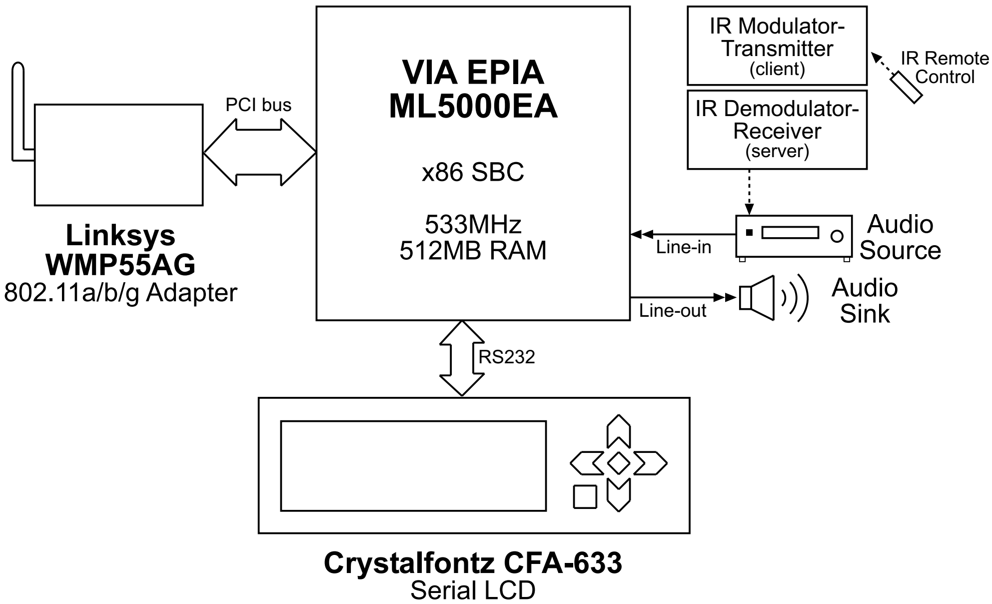

Figure 3.1: Hardware interconnection diagram

3.2.1 Single Board Computer

After exploring several options for the core of our hardware platform, we chose to use an x86-compatible single board computer (SBC). The widespread use of the x86 architecture allowed for many options when choosing the additional hardware as well as software to enable the functionality of the system. In addition, x86-based SBCs are available in many form-factors with varying types of on-board peripherals and standard interfaces for connecting additional peripherals.

The three major criteria for selecting the chosen x86-based SBC were size, included peripherals, and cost. A minimal size was desired as it would allow the proof-of-concept to be closer in size to a production model. Since the system needed to handle audio and interface with an 802.11a/b/g network, an audio codec and 802.11a/b/g wireless network capability were required peripherals. The total cost of each system needed to be kept to a minimum to allow completion of the project within the specified budget.

The SBC which best met all three criteria was the VIA ML5000 Mini-ITX SBC, a picture of which is provided in Figure 3.2. This SBC included a 500MHz x86-compatible processor, an onboard AC97 audio codec, and a PCI slot which allowed connection of an 802.11a/b/g network card. The size of the SBC was also small enough (17cm by 17cm) that the size of the proof-of-concept could be on the same order as that of a potential consumer product. The price of the SBC was low enough that it was not restrictive.

Figure 3.2: VIA EPIA-ML5000EA motherboard[14]

The chosen SBC utilized a standard DRAM interface for system memory and an IDE interface for a hard drive. A 512MB DRAM chip was selected as it would provide a significant overhead of memory for development and operation of the system. A standard IDE hard drive was used to store the necessary persistent data. Keyboard and monitor support allowed us to develop software entirely on the systems themselves. Also, debugging information could be displayed on the console which increased the ease and efficiency of the development process.

3.2.2 Wireless Network Adapter

It was decided that the system should be able to operate using 802.11a, 802.11b, and 802.11g so that the relative performance of each protocol could be assessed. Therefore, the wireless network adapter used needed to support all three protocols. The wireless network adapter also needed to be compatible with the hardware and software platforms.

The Linksys WMP55AG adapter, shown in figure 3.3, was chosen to satisfy these requirements. The WMP55AG supports 802.11a/b/g, has a PCI interface, and has Linux driver support. While there were several other wireless network adapters that satisfied these requirements, Linksys has established itself as a reputable brand. This indicates that their products are less likely to have design or manufacturing flaws, product support should be readily available, and the software drivers available should be fully functional.

Figure 3.3: Linksys WMP55AG wireless network adapter[15]

3.2.3 User Interface

A user interface (UI) was necessary to allow a user to operate our system. The UI had to provide the user with adequate control of the system as well as provide feedback on the current state of the system. In addition, the UI had to be compact in size to allow it to be integrated with the mechanical enclosure.

An LCD was chosen to provide system status information. LCDs are available in a variety of form-factors at low cost. The Crystalfontz 633 series LCD, shown in figure 3.4, was chosen. This display allows two lines of text containing sixteen characters each to be shown. The selection of the 633 series yielded a balance between the number of characters the LCD could display and cost. It also allowed selection of an LCD that could be mounted on the front of the chosen enclosure.

The 633 series also has six buttons that can be read by a computer. This simplified the design of the UI since it eliminated the need for a separate circuit to accept user input. An RS-232 interface is used by the 633 series to accept commands to display text and report button status. Use of this

Figure 3.4: Crystalfontz 633 LCD module[16]

standard interface ensured that the LCD would be able to communicate with other components of our hardware platform.

3.2.4 Infrared Repeater

The wireless communication between the audio sources and sinks needs to be bidirectional. Audio signals must be sent from the sources to the sinks via a ‘downlink’ and control signals must be sent back to the sources via an ‘uplink’. The downlink utilizes IEEE 802.11a/b/g because of throughput requirements of uncompressed audio. While the uplink could also be implemented using the Wi-Fi network, the extra traffic on the network could degrade the performance of the downlink especially when multiple transmitters are streaming. In anticipation of this effect, we decided to design separate radio frequency (RF) hardware to handle the control uplink. In this way, the IR repeater is completely decoupled from the audio hardware. This tactic also allowed us to simplify our code and hardware interface, mitigating difficulties associated with conversion of IR signals to and from a computer-readable format.

Because the IR remote signals do not require a high throughput, we had much flexibility in choosing the frequency of operation as well as the specific hardware for our implementation. Recall form §2.3 that there are a number of frequency bands available that offer the bandwidth necessary to send our control signals. We elected to use the 433 MHz frequency band. This band offers a combination of sufficient range ($\geq$50 m) and a small antenna, both of which were critical to the functionality and implementation of the devices. Once we had decided to operate around 433 MHz, we began researching wireless modules capable of transporting the IR-encoded binary information. We decided to use RF chips from Linx Technologies, as they are very low cost and also offer ultra compact, matched antennas. The dimensions of these antennas were only 1.1” by 0.54” by 0.06”, allowing us to avoid the addition of a second external whip antenna (like the one used for our Wi-Fi connection). Finally, they provided the capability to operate in a network. This is possible because the transmitters and receivers all operate on the same sub-band of the 433 MHz frequency; any receiver will pick up data from any transmitter.

The IR transmitter and receiver circuits are shown in figure 3.5. The detector device will receive the IR signal, demodulate it, and transmit it through our RF components. The emitter device will then remodulate it and send the signal to the audio source. We chose this approach, as opposed to just transmitting the modulated signal, for two reasons. First, demodulating down to a baseband digital signal allowed us to use RF circuitry that could operate on a bitstream. This allowed us to choose simpler, less expensive RF circuitry. Second, demodulating the signal allowed us to use high-quality IR receivers, much like those found in most audio equipment that receives IR signals. These IR receivers can receive the remote signal over a large range of distance, even when the remote control is not pointed directly at the IR receiver. Our IR repeater hardware is very compact, relatively inexpensive (approximately $25 per unit), and easily fits within the enclosure that was chosen for our other hardware.

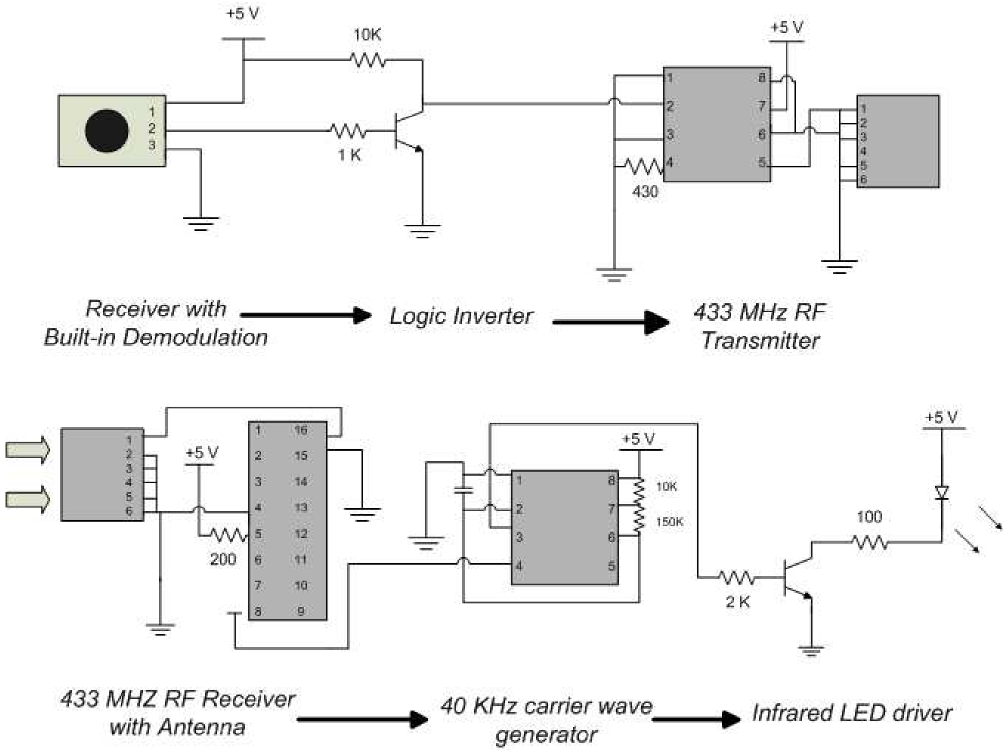

Figure 3.5: IR repeater transmitter and receiver circuits

The final step in the design of our IR repeater involved the creation of a means to transmit the IR signal from the RF receiver back into the audio source. While an IR light emitting diode (LED) is capable of performing this task, it needs to be placed in direct line of sight with the audio source. For this purpose, we built a wire that connects to our product and has an IR LED at the end of it. With this wire (seen in figure 3.6), we could position the IR LED very close to the audio source to ensure that the IR light reaches the source. For connection between the wire and our product we chose to use RCA connectors. We used these because they were capable of passing the type of signal we used through a shielded conductor and also because our product already had the space for an RCA connector. Additionally, these RCA connectors were available in bulk-head style, which allowed us to mount them onto our enclosures and provide simple, effective strain relief from the connections to our soldered circuit boards. The use of an RCA jack, as opposed to a permanent wire, has two major benefits. First, if for some reason the wire gets pulled on, it will simply disconnect from the enclosure without damaging the wire or the circuitry. Also if the audio source does not have an IR remote, the IR LED wire can be removed completely.

Figure 3.6: IR LED with RCA connector

3.2.5 Enclosure

Since the motherboard used was a Mini-ITX form-factor, we were able to select an enclosure with standard mounting points for Mini-ITX motherboards. The enclosure selected was the Casetronic ITX-2699R. This enclosure had mounting points and brackets to attach the Mini-ITX motherboard, the hard drive, and wireless network adapter. The enclosure also included an external 120VAC to 12VDC power supply and an internal DC to DC converter to supply the necessary voltages to the internal components including the motherboard, the IR repeater, and the LCD display.

Figure 3.7: Inside enclosure view

The enclosure required modification to accommodate both the LCD module and the IR components (emitter and detector) on the front panel. The front part of the case was removed and cut at the Washburn machine shop at WPI. A large rectangular hole and four mounting screw holes were cut out of the center of the piece to accommodate the LCD board. Additionally, a small circular hole was cut out of one of the unused drive bay covers on each machine for our IR components. On the client machines, a small IR light filtering plastic material sits directly behind the hole, behind which the IR detector is mounted. On the server machines, an RCA female jack is mounted to the front of the enclosure, to which an RCA male terminated IR LED can be connected. Photos of the mentioned case modifications are available in figure 3.8.

Hardware Integration

Figures 3.8 and 3.9 show the final mechanical layout of all of our hardware components. The non-standard computer parts, namely the LCD and our infrared repeater circuit boards, are mounted outside of the metal chassis to the plastic front panel of the enclosure.

Figure 3.8: External view

3.3 Software

This section will describe the major functionality of our system and describe in detail the software that achieves this functionality. We wrote all of our programs in C on the Linux operating system. [1]

3.3.1 Operating System

One of the major advantages of using an x86-based hardware platform was the wealth of software available to abstract the hardware layer and allow development of applications in high-level languages. There are several operating systems that can be used to accomplish this. Debian GNU/Linux was chosen because it offers many benefits for this application. The design of the operating system provides a simple interface to network and sound devices, both of which are essential components of the system. There are also many resources for developers programming for these devices.

[1] Readers not familiar with the C programming language and specifically Unix interprocess and network communications may benefit from referring to a good C manual such as the free GNU C Library, available at `http://www.gnu.org/software/libc/`

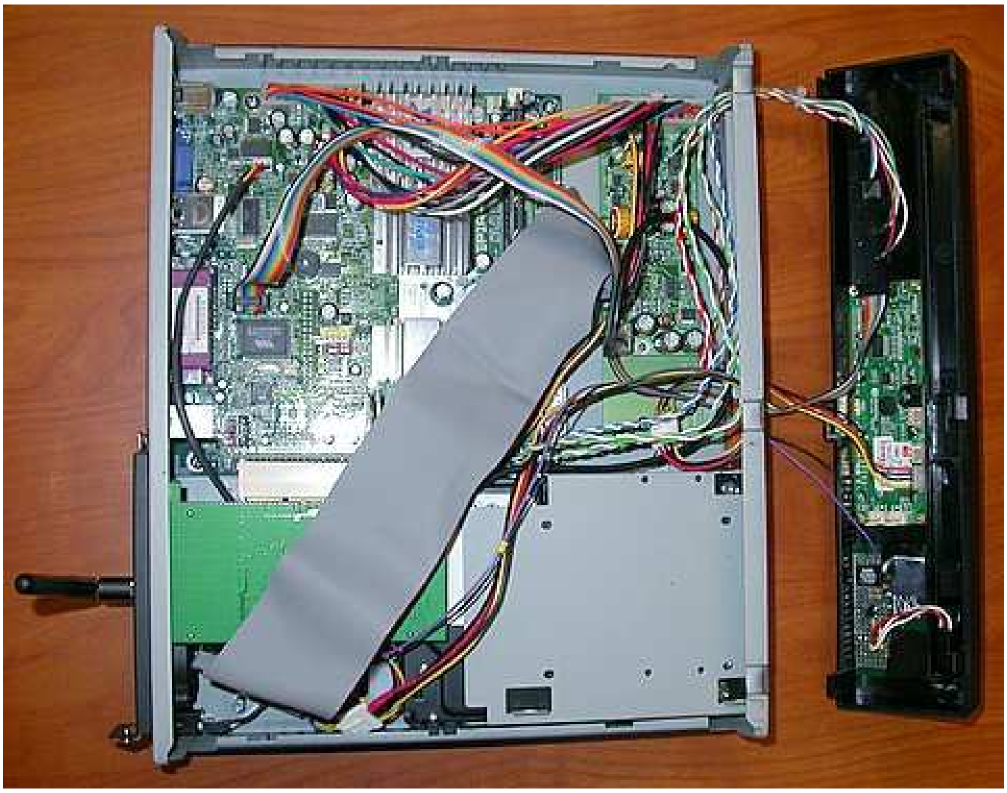

Figure 3.9: Internal view - top and front cover removed

Another advantage of Linux is its stability; Linux is known for not exhibiting undesirable behavior such as inconsistent performance or “locking up.” This is an important feature since the operating system influences the performance of the entire system. Linux is also free which reduces the overall cost of each system.

3.3.2 Software Design

The software for our wireless music distribution system was divided into high-level functional components. This section will describe the functionality these components to allow the reader to understand the purpose of each component of our code and understand how they interact. For the purpose of describing our software, we will define the transmitter units of the system as *servers* and the receiver units as *clients*. This terminology reflects the functionality of our units in the context of network communications. Our system is designed to be expandable to support many servers and many clients. However, we will consider a 3-by-3 system, as shown in figure 3.10, to illustrate it's operation. Three servers and three clients are shown as boxes on the left and right sides. The circles in the center are multicast IP addresses which data can be sent to or received from by any unit.

Audio Streaming

One of the most basic functional requirements of the system is the ability to stream music. The uncompressed audio streams that our system supports require much more throughput (1.4Mbps

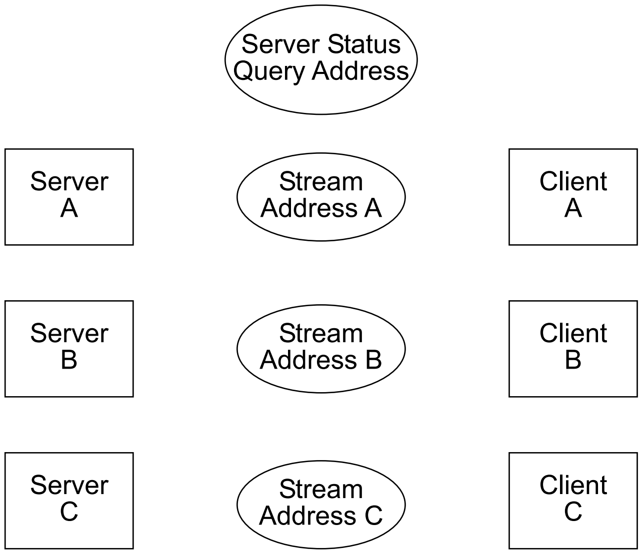

Figure 3.10: Network of servers and clients

as determined in §2.2) than do compressed audio streams, which is what many of the previously mentioned Wi-Fi based products support. Furthermore, the system must be "many-to-many" capable; each of the severs must be able to stream to any number of clients.

Consider a situation when m clients wish to receive streamed music from a single server. If the streams were sent using TCP/IP unicast, the server would need to send separate streams to each of the m clients. The network throughput requirement would then be proportional to m. In general, the total network throughput requirement of an n server by m client system would be proportional to the number of active clients, with a maximum of m. This is an inefficient use of bandwidth as much of the network traffic is redundant.

To aid in reducing the throughput required of the 802.11 network, our method employs UDP/IP multicast streams. Each server streams to a multicast IP address which clients can "tune in" to. This is analogous to FM radio stations: the station broadcasts on a single frequency and any number of radios can tune in to this transmission. Streaming audio via multicast ensures that no redundant information is sent. To further reduce throughput, we have designed our system such that servers are aware of the number of clients tuned in to them at any time. This way, servers that have no client connected to them will stop transmitting to conserve network throughput. For this system, the total network throughput requirement is proportional to the number of active servers, with a maximum of n. Since the number of active servers is always less than m, this configuration is more efficient than a unicast system.

Figure 3.11 demonstrates a situation where two clients are listening to a stream from server A while a third client is listening to a stream from server B. Sever C has no clients connected to it, so it is not transmitting audio packets in order to conserve network bandwidth.

Server Discovery

When a client joins the network, it needs to be able to discover what servers exist on the network. This is accomplished by using a *Server Status Query multicast address*. This address is known to

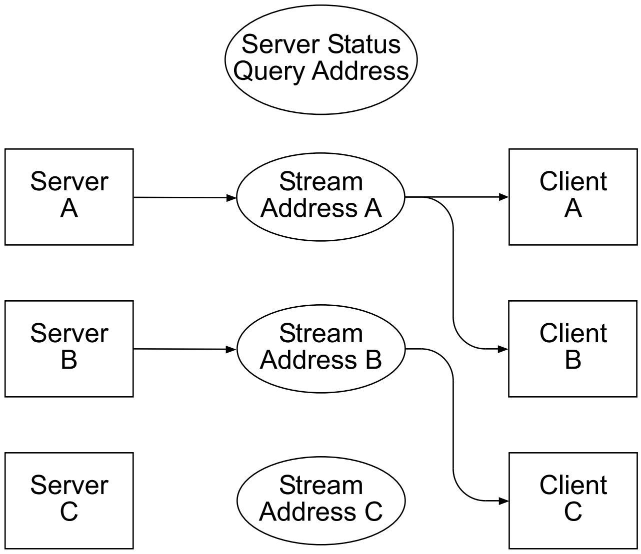

Figure 3.11: Audio streaming over the network

all clients and servers before connecting to the network. When a client joins the network, it sends a request for status information to this address. All servers on the network listen to this address and respond to the client using a TCP/IP unicast connection. The response includes the server's logical name, the server's IP address, and the server's stream multicast address. After the query and response operation has completed, the client has a complete list of all available servers on the network. An illustration of this operation is presented in figure 3.12.

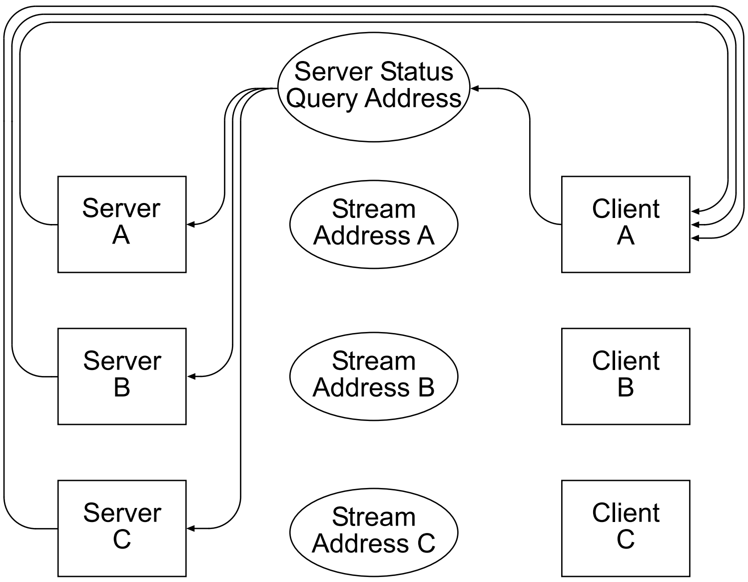

Figure 3.12: Server Discovery initiated by client's query

Control and Status Communications

Servers and clients that are connected need to be able to send information to one another. For instance, a client sends a signal to a server whenever it connects to or disconnects from that server's stream address so that the server can keep a count of how many clients are connected. Since these messages are critical, they are sent via a TCP/IP unicast connection, as shown in figure 3.13. Because TCP/IP unicast employs packet confirmation and retransmission, the messages are sure to reach their destinations. While these messages require relatively more network bandwidth because of the overhead associated with this handshaking, the messages are small and relatively infrequent so the additional load on the channel is acceptable.

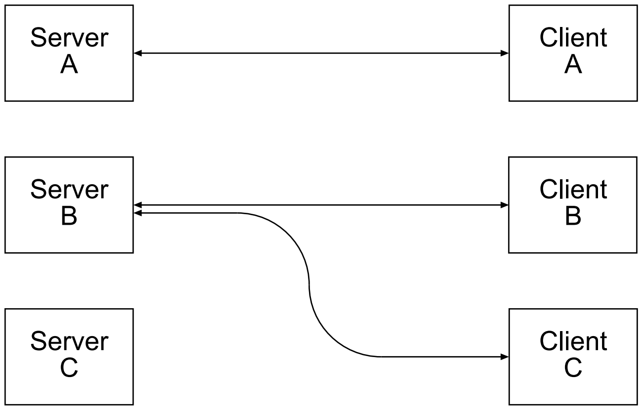

Figure 3.13: Control and status communications between clients and servers

3.3.3 Software Implementation

We will now describe the programs we have written to perform each of these functions. We elected to develop our software as separate programs that run concurrently to allow for parallel execution of independent tasks. The majority of our functions were implemented with server-client pairs of applications with the exception of the user interface which runs only on the client machine.

The server applications and their operation are shown in the block diagram below, figure 3.14. Similarly, the client applications are shown in figure 3.15. The solid arrows indicate a process initiating another process (for example the Control and Status Handler initiating the Query Handler). The dotted arrows indicate inter-process communications, either by means of a data pipe or software interrupt (a.k.a. signal) and the dashed arrows indicate network communications between client and server machines.

Streamer and Player Applications

The Streamer and Player applications perform the audio transport function of the system. On the server side, the Streamer program, `streamer`, reads audio samples from the sound card. In Linux, this is achieved by reading from the */dev/dsp* device. After 1024 bytes (or 256 stereo 16 bit samples) have been acquired, they are sent in a multicast packet to the server's designated stream address. This packet size was chosen because reading from the sound card is most efficient when

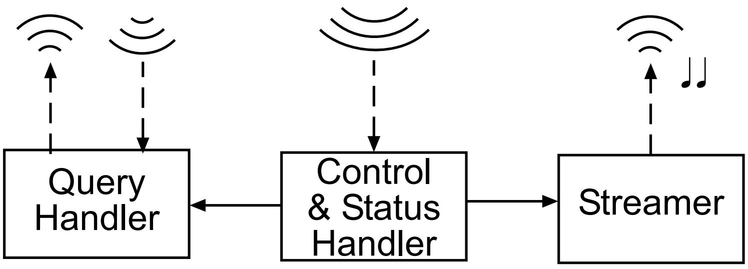

Figure 3.14: Server Applications

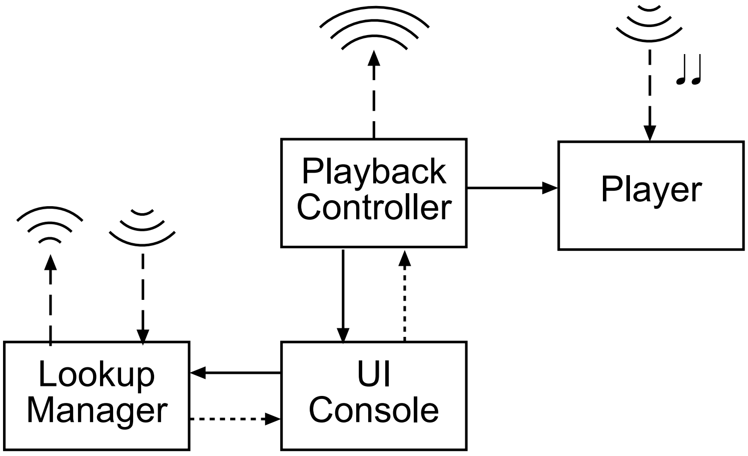

Figure 3.15: Client Applications

it is done in block sizes that are powers of 2. Since the MTU (Maximum Transmission Unit) for a datagram is 1500 bytes, a block size of 1024 ensures a minimal amount of average packet overhead. The operation is looped indefinitely until the `streamer` process is killed.

The complimentary client application is the Player program, `player`. When `player` is initiated, it is given the desired server's stream address as a command line parameter. It receives packets from this address and writes them to the */dev/dsp* device, which causes the audio to be sent to the line out of the sound card. Like `streamer`, `player` will continuously loop until it is killed.

Lookup Manager and Query Handler Applications

The server discovery operation is implemented with a pair of programs, the Lookup Manager on the client side and the Query Handler on the server side. When the Lookup Manager application `lookup` is initiated, it makes a connection to the server status query multicast address. It sends a packet containing it's own IP address and port for TCP/IP communications to this address. Since the UDP multicast packet does have guaranteed delivery, `lookup` sends this same information five times. The intent of this redundancy is to increase the probability that the message is received by the servers. After `lookup` has sent it's information to the Server Status Query address, it listens for replies from servers on the port specified in the sent query message.

The Query Handler application, `query` runs constantly on the server machines. `Query` listens to the Server Status Query multicast address, waiting for client queries. When it receives a packet, it parses the message into the IP address and port and saves these values in local variables. It then forks off a child process to handle the reply so that the parent can listen for more queries. Each child process establishes a TCP/IP connection to the IP address and on the port specified in the message. To that IP and port, query sends the server machine's information, including the unit's name, it's IP address and a port for TCP/IP communications, and the server's multicast stream address and port to listen on. Since this message is sent via TCP/IP, it has guaranteed delivery and therefore does not need to be sent redundantly. One feature of the Query Handler is that for a given amount of time after a first query, it only responds to unique IP addresses. This prevents unnecessary redundant replies to the same client; the client will query five times but will only be responded to once. After a short period of time, query clears the list of 'old' IP addresses so that clients can perform server lookup operations again at a later time without being ignored.

As stated before, the Lookup Manager waits for server replies after it has sent the client machine's information. Similar to the query, lookup forks a new child process to handle replies every time a new server responds. This way, if server A and server B both respond to a client's query at approximately the same time, the client will not miss the second response while it is processing the first. Each child process establishes a TCP connection with the responding server and receives the server's information. It then parses this message, stores the server's information in a file, servlist.txt, and quits. When ample time has elapsed following the initial query, lookup stops listening for server replies and quits. By this time, the servlist.txt file contains a complete list of all servers on the network.

Control and Status Handler and Playback Controller Applications

The Control and Status Handler, control, and Playback Controller, playback, applications are the main threads from which all others spawn on the server and client machines respectively. When the machine is started up, one of these two programs is executed and the machine goes into either server mode or client mode.

When the Control and Status Handler program is run, it spawns a Query Handler process immediately. Query runs in the background continuously while the machine is in server mode. Control also handles TCP/IP communications with clients. It will bind to a designated port and wait to accept incoming messages from clients. Like many of our other programs, when a client connects, control spawns a child process to receive and handle the message while the parent waits for more connections. A client will either send either a 'connect' or a 'disconnect' message to a server which will react by incrementing or decrementing a counter variable that keeps track of the number of listening clients. If that counter becomes greater than zero, control will initiate a streamer which will start streaming audio to that server's designated stream address. If the counter becomes zero, the streamer process is killed, preserving network bandwidth.

The Playback Controller initiates the user interface program on startup using a *popen()* command. This opens a one-way pipe between the child process's standard output (*stdout*) and an input file descriptor in the parent. The UI program sends 'play' and 'stop' commands to the Playback Controller using this pipe. A 'play' command comes through as an integer corresponding to a server in the servlist.txt file. The playback controller opens this file and extracts the necessary information for the requested server. It then sends a 'connect' command to the server via TCP/IP, and initiates a player with the server's corresponding multicast stream address. When playback receives a 'stop' command, it sends a 'disconnect' command to the server and kills the player program. If the client is already playing a stream from one server and another 'play' command appears on the pipe, `playback` performs the 'stop' operation, followed by the 'play' operation so that there are never multiple `players` active at any time.

3.3.4 User Interface Design

The UI was designed and implemented as a series of states. Each state consisted of a message to be displayed on the LCD and a defined set of responses to button presses. These states can be seen in table 3.1. States 10 through 60 represent the Player Mode (client operation) and state 500 represents the Source Mode (server operation).

The button press responses were designed to be as consistent as possible between states. For example, the **LEFT** button switches between Player and Source Modes, the **RIGHT** button searches for additional music sources in many of the Player states, and the **CHECK** button plays the selected source in several modes. This was intended to increase usability as it reduces the number of functions each button has.

The text displayed was written to be succinct so that it would fit into the 32 character spaces available on the display. This necessitated careful consideration to choose the most important information to display in any given state.

3.4 System Performance Testing

After the hardware and software was designed and integrated, it was necessary to test the system performance to ensure it met the technical specifications and to find ways of improving the final product. Performance tests were designed to test the audio streaming capabilities of the system. The first test determined the throughput capacity of each system. The throughput determines the maximum rate at which each device can transfer information. The second test measures how the packet loss increases with distance. This is directly related to the audio quality over distance. The last test measured the effect of interference caused by simultaneous audio streams.

3.4.1 Throughput Testing

To test the throughput of the wireless link, a program was written to multicast UDP packets at various rates. The test program multicasts the packets with the exact same method as the streamer application in the software design. The test program allows the tester to specify how many packets are sent at once (burst size), the delay between packet bursts, and the packet size. Another test program listens on the multicast address and port and records each of the packets it receives to a file. Each packet is numbered by the transmitting test program so the receiving program can scan its received packets file to locate missing packets. It then records the total number of packets lost and calculates the packet loss rate. The transmitting and receiving test programs are called `streamtest` and `streamrecv` respectively.

These programs were used to test various throughput rates by adjusting the burst size, packet size, and burst delay parameters. The approximate throughput in bytes per sec can be calculated simply by multiplying the burst size by the packet size and dividing by the burst delay. Initial testing on laptop computers showed that independent parameters had little effect beyond the total throughput as long as the packet size was less than the Maximum Transmission Unit (MTU)

Table 3.1: UI state transition table

		Button Press response (state # or action)					
State #	**Display Text**	**UP**	**DOWN**	**LEFT**	**RIGHT**	**CHECK**	**X**
10	`****ORPHEUS*****` `Press a button!`	20	20	20	20	20	20
20	`UP:Player Mode` `DOWN:Source Mode`	30	500	500			
30	`Searching for` `Music sources...`			500			
40	`Choose source:` [Available Server Name]	Display previous available server name	Display next available server name	500	30	Set [Current Server Name], 50	
50	`Now playing:` [Current Server Name]			500	30		60
60	`Ready to play:` [Current Server Name]			500	30	50	
500	[Name of Self] `Listeners:`[# of listeners]			30			Previous state

(1500 bytes), the burst size was less than the total transmission buffer, and the burst delay was greater than the resolution of the Linux system timer (around 1 ms).

The throughput capacity was determined by measuring the packet loss as the data transmission rate was increased by raising the packet size or burst size or lowering the burst delay. The packet loss typically stayed fairly constant at or below 1% until the throughput capacity was reached and the packet loss increased rapidly. The results of the throughput testing are given in §4.1.

3.4.2 Range Testing

Another test program was developed to measure the wireless performance of transmitting an audio stream over distance. The program, called alsatestplayer, is a modified version of the player application used for receiving and playing the wireless audio stream. The application receives 5200 packets which correspond to about 30 seconds of PCM audio from the audio streamer application. It saves the audio stream to a wave (**.wav*) file and keeps a counter of each missed packet detected from the packet numbering. The number of missed packets is saved to a file for the trial. The wave file was used to subjectively assess the effect of packet loss on audio quality. During testing, a script was used to run the program ten times consecutively for approximately five minutes at different distances. The results of these tests are reported in §4.2.

3.4.3 Self-Interference Testing

Tests were conducted that measured the effect of simultaneous audio streams on packet loss. In the absence of any other wireless networks, two server machines transmitted audio streams at a distance of 30 meters from a client machine that was set to receive one of the streams. Using the alsatestplayer application, the audio stream was recorded and packet loss measured. This test setup is shown in figure 3.16.

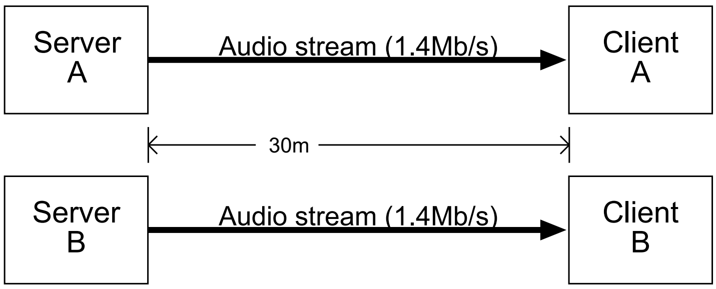

Figure 3.16: Unidirectional self-interference test setup

Another test was conducted in which one audio stream was transmitted one direction while another audio stream was sent in the opposite direction at a distance of 30 meters. This configuration is shown in figure 3.17. One of the client machines used alsatestplayer to record the audio stream and measure its packet loss. The client machine was subject to greater interference than the first test because it was trying to listen to the server 30 meters away while another server directly next to it was transmitting a different audio stream. A single audio stream at 30 meters was used as the control for these experiments. The results of the self-interference tests are included in §4.3.

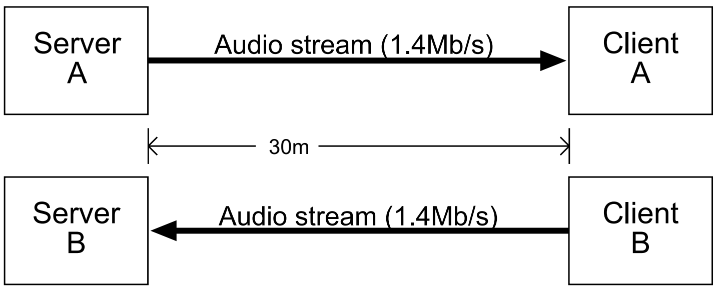

Figure 3.17: Bidirectional self-interference test setup

Chapter 4

Results

The major result of this project was a functional two-by-two proof-of-concept digital wireless music distribution system. This system had all the desired product features such as lossless streaming of digital audio, compatibility with generic sources, an intuitive user interface, and source control implemented by an IR repeater. It was necessary to test the proof-of-concept system in order to measure its performance and ensure it met all the technical specifications determined in §3.1.

Wireless tests were performed following the methodology described in §3.4. These tests were designed to characterize the throughput, range, and scalability of the many-to-many wireless music distribution system. This chapter will present the results of this testing and the conclusions reached from them.

4.1 Throughput Test Results

Using the procedure explained in §3.4.1, the maximum throughput was determined for an 802.11a and 802.11g wireless link. The data rate of the transmission was slowly increased by reducing the time between packets until packet loss rapidly increased. Figure 4.1 shows the results of these tests. The graph shows a steep rise in packet loss for 802.11a at around 5.5 Mbit/s and 11 Mbit/s for 802.11g. The steep rise can be attributed to reaching the maximum throughput capabilities of the wireless hardware.

While the maximum theoretical throughput of these wireless protocols is 54 Mbit/s, there are many factors that limit the throughput that can be achieved in practice. There is overhead associated with transport protocols like UDP and Media Access Control (MAC) protocols like CSMA/CA. The native Linux driver used could also not be completely optimized for each of the network links. It is important to recognize that the throughput limits determined do not represent the limit of the entire wireless channel. This was confirmed by streaming from two systems at close to their maximum throughput and measuring packet loss. The packet loss increased noticeably but did not hit a throughput wall that would indicate the limit of the entire wireless channel was reached. The packet loss increase in this case appears to be due to the network interference and crosstalk from simultaneous streams. This is important because it shows that while one device may be limited to transmitting at 11 Mbit on 802.11g, two or more devices can still transmit at 11 Mbit simultaneously without reaching the total channel limit for RF transmission.

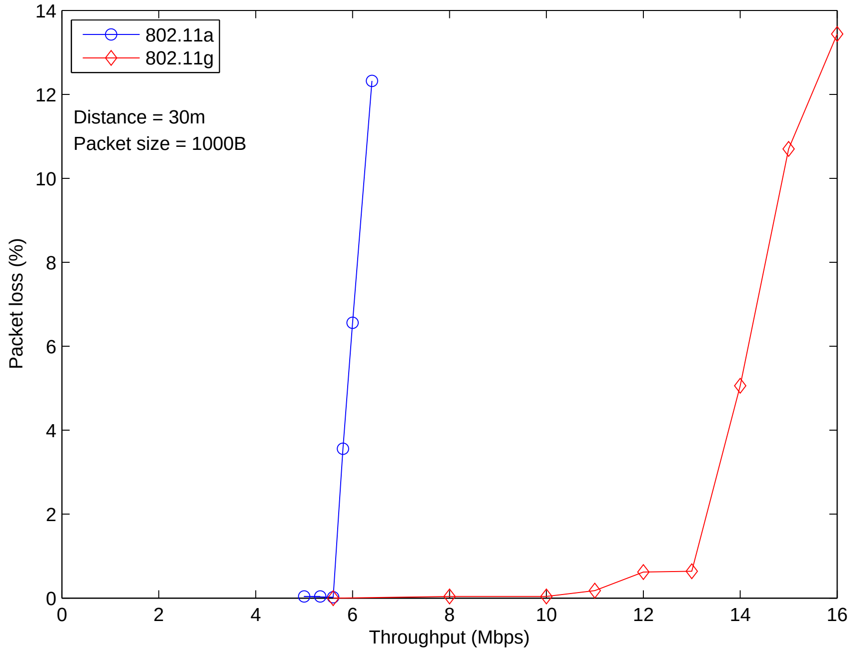

Figure 4.1: Packet loss vs. throughput

4.2 Range Test Results

The methodology described in §3.4.2 was used to conduct range testing to find how the system scales with distance. 5200 packets corresponding to about 30 seconds of audio were streamed at distances from 5 m to 45 m. Each trial was repeated ten times. Figure 4.2 shows the packet loss over distance results of an 802.11g link. The bars in the graph represent the standard deviation of the measured samples. The graph indicates a nearly linear increase in packet loss over distance with line-of-sight. Even at 45 meters, the packet loss is well below 1% of the total transmitted packets.

The 802.11a link was able to achieve 0% packet loss at 45 meters with line-of-sight for a 30 second audio sample (5200 packets). This shows 802.11a's major advantage over 802.11g in these tests. While 802.11a was limited to a throughput of only 5.5 Mbit, it had significantly lower packet loss than 802.11g over distance. This appears to be due to the OFDM technique employed by 802.11a described in §2.3.3.

The IR repeater was also tested under the same conditions. Not surprisingly, the presence or lack of other 802.11 wireless networks had no effect, as the repeater operates in a different frequency band. In range tests, the IR repeater proved to function very well at 45 meters in line-of-sight. Multiple receivers were also able to receive the signal from a single transmitter simultaneously at this range. When multiple transmitters were broadcasting simultaneously, there was a chance of the transmission from either of these being missed. However, because these devices are occasional transmitters, the odds of an overlap in transmission would be very low, and could be easily remedied by the user simply repressing the remote control command.

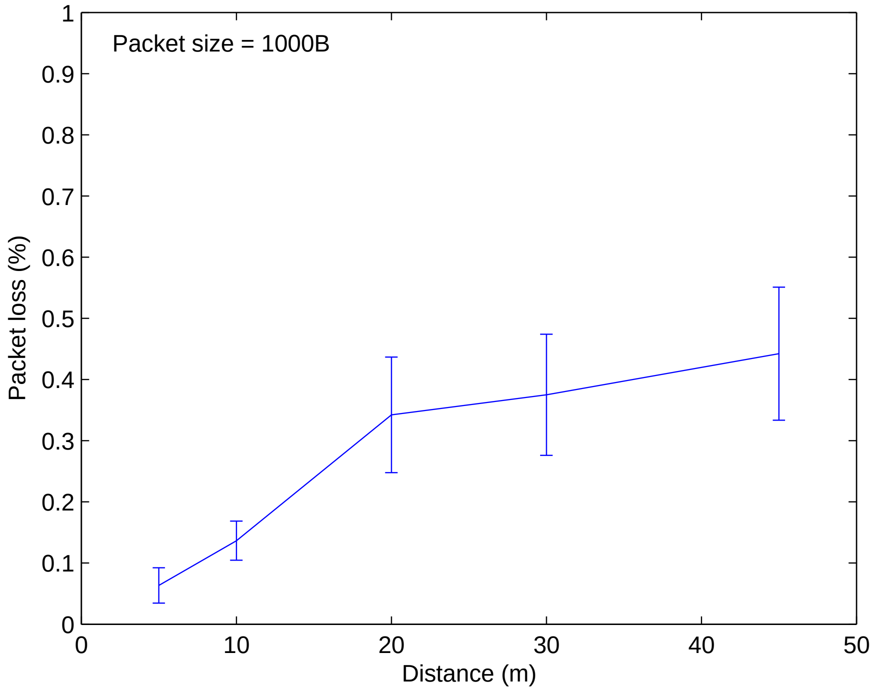

Figure 4.2: 802.11g Packet loss vs. distance

Table 4.1: Self-interference test results

Test	Packet Loss 802.11a	Packet Loss 802.11g
Test A (1 to 1)	0%	0.05%
Test B (Unidirectional)	0.13%	0.15%
Test C (Bidirectional)	0.22%	0.25%

4.3 Self-Interference Test Results

In order to quantify how the system scaled with multiple streams, tests were performed using the procedure described in §3.4.3. In the one to one case, a single audio stream was transmitted at 30 m. The unidirectional test sent two streams simultaneously from two servers to two clients at a distance of 30 m. The bidirectional test reversed the direction of one audio stream so each listening client was below a transmitting server. The results of these tests are presented in table 4.1.

The table shows how packet loss increased when subject to more interference from simultaneous streams. The system scales reasonably well with two simultaneous streams. As expected, the bidirectional case introduced the most interference because each client was immediately below a streaming server while attempting to listen to another audio stream being sent 30 m away. However, due to the collision detection management capabilities of 802.11 even in this case the packet loss was well less than 1%. Unfortunately, it was not possible to test with more streams because only a two-by-two system was implemented.

Results Summary

The testing performed demonstrated the single device throughput limit, the wireless range, and the scalability with simultaneous streams. 802.11g had about twice the throughput capability as 802.11a for a single device. This could become important if a larger data stream is used such as video. The exact cause of the throughput limit could not be determined, however, and could simply be the result of a non-optimized driver that is still under development.

802.11a demonstrated better scaling with distance and achieved packet loss free streaming at 45 meters. 802.11g also performed well with packet loss rates under 1% up to 45 meters but there was a noticeable increase with distance.

Both wireless protocols performed similarly with self-interference caused by simultaneous streams. The tests demonstrated that the two-by-two system performs well under worst case conditions for self-interference. Further testing remains to characterize the system with more devices and under external interference.

Chapter 5

Conclusion

As technology evolves, increasingly elegant and practical solutions to consumer needs become possible. Recently, advances in data storage and IC design have enabled the proliferation of a wide range of digital music storage and playback devices. A particularly strong demand exists for devices which allow users increased mobility while accessing personal music libraries. The unparalleled success of the iPod is one of the most visible examples of this demand.

While there has been a focus on the development of portable audio products, companies are only recently beginning to create products which allow consumers access to music throughout a home. Despite a recent boom in home-wide music distribution systems, currently available products fail to meet all consumer needs.

This project identified a set of features for which consumer need is visible that have not been implemented on any currently available product. This feature set allows consumers to use their existing audio equipment and music libraries while enjoying their music at any desired location in their home. In contrast to traditional systems, this functionality is achieved without the need to install costly or inconvenient communication infrastructure. Furthermore, the feature set allows multiple users to access the system simultaneously from various locations throughout a household, ensuring that the need for a truly home-wide music distribution system is met.

To demonstrate the feasibility of such a feature set, this project developed and implemented a proof-of-concept system. The system allows one to experience the convenience that such a feature set affords the user as well as illustrating methods of utilizing available technology to enable a wide range of functionality.

In order to allow users to not only listen to but control audio sources remotely, an IR repeater was developed. This repeater allowed users to utilize IR remote-controls supplied with audio products from any location in a home where the system is in use. Care was taken to ensure that the IR repeater was interoperable with all commonly available IR remote-controls.

The proof-of-concept system utilized a digital wireless link to transport music between audio sources and listening systems. Due to congestion of the RF spectrum, the throughput of this link is restricted. In order to effectively utilize the available throughput, care was taken to ensure that the transmission of redundant or unnecessary data was minimized. This led to a system design which simultaneously allowed uncompromised audio quality and efficient wireless spectrum usage.

The system is expected to be highly scalable. If the system were able to efficiently use the throughput offered by 802.11a/g, 38 simultaneous streams of uncompressed audio would be possible. However, initial scalability testing indicated that the actual number of streams which can be concurrently transmitted is less than this. Additional testing and investigation of the 802.11a/g protocol could yield the maximum number of simultaneous streams. Techniques such as lossless compression could further increase this number.

One of the significant factors in determining the performance of the system was the quality of the wireless link. In order to characterize the performance of the system, thorough testing of the wireless link was performed under varying conditions. This testing revealed a range of factors which yielded ideal performance of the system as well as conditions that can potentially cause undesirable operation. We found that packets lost during transmission, while often a small percentage of the total packets sent, were still a major problem. Additionally, because our prototypes were built using single board computer platforms, the size and unit price are not feasible for commercial production. These problems led to the creation of a number of recommendations (found in §6)which can be used to improve the performance and marketability of the system.

Should the system be further refined and brought into production, there is potential for commercial success. Consumer trends indicate the need for the features implemented in this system. The availability of off-the-shelf components to enable functionality of the system ensure that the complete system can be supplied to consumers at a reasonable price. Furthermore, the recent boom in wireless home-wide audio solutions has created the consumer awareness necessary to facilitate rapid market adoption.

Chapter 6

Opportunities for Further Development

While we achieved our objectives of designing a many-to-many digital wireless audio system and building a two-by-two proof of concept, many of the design decisions we made were limited by the resources we had available to us. Given more time and a larger budget, there were several features and improvements we would have liked to implement. We have investigated and documented these ideas, providing a potential starting point for future research and development.

6.1 General Product Improvements

The decision to implement the system x86 SBCs was made early in the project with the intent of reducing the potential difficulties associated with the integration of external hardware (such as the Wi-Fi cards and LCD displays). An offset to the convenience of these SBCs, however, is their cost. From a commercial stand-point, the use of this hardware would almost certainly be too expensive. The per-unit cost would greatly exceed that of the prior art that we evaluated. Therefore, it would be much more practical, to implement the system on less expensive hardware with minimum resources (such as memory, storage, and processing power).

Another major decision we made involved audio compression. In accordance with our project goals, it is imperative that any compression and decompression would need to be both real-time, and lossless. We determined that, given the time constraints on our project, it would be infeasible to design and implement any such compression. Should a compression scheme be implemented, it could increase the overall scalability of our system by lowering the bandwidth required by each audio stream. The maximum compression achieved by lossless compression schemes is approximately 50%, but can range from 35% to 90% depending on the dynamic range of the audio. Assuming that the system is linearly scalable, this would result in an approximate doubling of the maximum number of concurrent streams. This would add only a small load to the CPU and would incur no additional hardware cost.

Our scalability estimates are based on a simple analysis of the available throughput of the channel divided by the throughput required for each uncompressed audio stream. Our test results, however, suggest that this simple analysis may be optimistic in that the interference caused by each stream reduces the available bandwidth by more than 1.4 Mbit/s.

6.2 Further Testing

There are a number of tests that could be conducted in the future to provide more information about our wireless system. Our scalability estimates are based on a simple analysis of the available throughput of the channel divided by the throughput required for each uncompressed audio stream. Our test results, however, suggest that this simple analysis may be optimistic in that the interference caused by each stream reduces the available bandwidth by more than 1.4 Mbit/s. We have been unable to run tests with more than three servers transmitting simultaneously and therefore do not have a full understanding of the channel limitations. A test to quantify the scalability of our system could involve adding more audio-streaming transmitters to the network until a true limit of the channel was determined. Results of this testing would not only reveal the scalability of our system but would also provide a basis for future improvements (such as the implementation of real-time compression) if the scalability was determined to be insufficient.

One other form of testing that could be very beneficial for future applications of our product, as well as other wireless technology, would be the isolation and characterization of interference. One issue with the results of our distance and throughput test was the environment in which the tests were conducted. We attempted to run these tests in the absence of any other wireless network or other stray interference around our transmission frequency. We found that, when not in this controlled environment, our results were much less repeatable. Because the use of wireless technology is growing very quickly, a better understanding of the sources of interference and how their signal strengths, positions, and data rates affect other products would be very useful. This test could potentially be very complicated, from both a testing and theoretical standpoint. While these issues were not addressed, we recognize them as very important to the implementation of wireless technology.

6.3 Mitigating Packet Loss

As described in the Results section, the UDP protocol for wireless transmission does not offer guaranteed packet delivery from source to client. Therefore, packets can be lost, resulting in degraded audio quality. While information was collected on the ranges and some environmental effects that cause packet loss, we were not able to implement any solutions to this problem. However, with more time, we believe there are several options for remedying or masking this problem.

One option for disguising packet loss is through the use of **redundant transmission**. In it's simplest form, this could refer to simply sending multiple redundant streams in hopes of successfully transmitting at least one of each necessary data packet. However, using intelligent coding techniques, data can be transmitted in such a way that the information contained in a lost packet can be reconstructed through data contained in other packets. This could allow for a certain percentage of packets to be lost without any degradation in audio quality without drastically increasing the throughput. The specific techniques to be used would need to be further researched and analyzed to determine their potential effectiveness in our system. This system could further be supplemented through the use of link status based throttling, where the strength of the wireless link would have an impact on how much the data between packets overlaps, and thus how many packets could be lost without a degradation in audio quality. This could potentially allow for minimal increases in the amount of additional throughput required in the presence of a strong link (with relatively low packet loss), and would also allow for minimal audio degradation in the presence of a very weak link.

Another potential solution to the problem of packet loss is the use of **selective packet retransmission**. This would rely on packet sequencing (like that currently implemented in our test programs) to alert the client when it has missed a packet. When a packet is missed, the client would send a message back to the server and the server would resend the missed packet. If another client listening to the same multicast address also receives this resent packet, it will recognize the redundancy and ignore it.

One final method for dealing with packet loss is through simple **data concealment**. Currently, when a packet is lost our software simply replays the preceding packet in place of this lost packet. However, doing so causes a discontinuity in the time domain, which in turns causes an audible 'pop' noise from the speaker. Data loss concealment techniques could be implemented to eliminate this popping noise. In one simple example of this type of technique, when a lost packet is detected, the final sample from the preceding packet and the first sample from the proceeding packet can be used to create a new packet. This packet would be of same length as the missed packet (to maintain the time integrity of the replayed audio sample) and could be a simple linear interpolation between the two nearest audio samples. This would eliminate the discontinuity, and thus the 'pop'. It could potentially create other unwanted audio effects, but this is an issue that would need to be investigated and tested further.

Summary

Following these recommendations could yield a music distribution system comparable in size and cost to currently available products. A packet loss handling method could improve the system's immunity to interference, allowing it to operate in higher noise environments and with multiple concurrent streams. Scalability testing would provide information on how the system behaves as the number of simultaneous streams increases. While these recommendations are provided as methods of improving our music distribution system, solutions to these problems could have a wide variety of applications. They could be useful in any system which requires streaming of real-time data between multiple nodes in a networked setting.

Appendix A

Project Description

Available at `http://spinlab.wpi.edu/Projects/opportunities/`

Many-to-Many Digital Music Distribution System
Project Sponsor: Bose, Inc.
Terms: A05-C06
Project description:

In this project, students will develop a many-to-many digital wireless music distribution system. In general, such a system would have M audio sources (e.g. a CD player, a DVD player, etc.) and N audio sinks (typically amplifiers and speakers but also wireless headphones, etc). No cables connect the sources to the sinks. The students will develop a transmitter module that uniquely identifies each audio source, allows for control of the audio source, allows for querying of the state of the audio source, and allows for digital audio streaming from the source. The students will also develop a receiver module that allows the user to select an audio source, request the state of the audio source, control the state of the audio source, request audio from a desired source, decode the audio, and play the audio. A key component of the project is that multiple sources should be able to stream audio simultaneously to multiple sinks. The general benefit would be that no cables would be required and you could listen to any audio source with any audio sink.

The student team will be required to research prior work in this area as well as build and demonstrate a 2 source by 2 sink proof of concept. The student team will evaluate and select existing digital wireless technologies to realize the wireless links.

Appendix B

Product Requirements

The work done on this project is based on the project description, shown in Appendix A. This description served as customer requirements, and was the basis for creating a list of minimum product requirements. Together, these minimum requirements are unmet by any product currently in the consumer market.

- **Wireless digital communication link**
  This requirement is fundamental to the project, with "wireless" and "digital" being two basic customer requirements
- **Many-to-many scalability**
  This means that while it is only necessary to physically build two transmitters and two receivers, the software designed and the hardware platforms selected must be capable of incorporating more wireless components into the system.
- **Meet Federal Communications Commission (FCC) standards for continuous wireless transmission**
  The FCC is the governing body in charge of all wireless transmission, and it is necessary for our product to abide by their regulations. Failure to adhere to these rules can result in serious fines, and would invalidate the project as a marketable product.
- **Ensure compatibility with home audio sources and sinks**
  One of the major shortcomings of current wireless audio products is the lack of compatibility with many common home audio sources (such as CD-players) and also home audio sinks (such as speakers and stereos). This product will address that shortcoming by having a common audio input format and audio output format to enable connection to almost all consumer audio equipment.
- **Provide source control from receiver**
  For usability purposes, it is very important to have some form of source control from the receiver module. A consumer listening to wirelessly transmitted music in one room will not want to have to go into the room with the audio source just to pause, play, fast-forward, etc. This product will therefore provide added convenience and usability by allowing the user to perform at least rudimentary functions from any room with a receiver in it.
- **Provide visual indicators for user**
  This requirement is again based on increasing simplicity and usability for the consumer. We

plan to have some form of user interface on the receiver module to show the existence of a wireless link between a transmitter and receiver, and also to indicate the reception of any commands that are being sent back to the source.

- **Provide home-wide transmission of audio signal**
  This requirement necessitates the selection of some wireless transmission technique by which receivers can be set a reasonable distance apart, with walls between the transmitter and receivers.

Appendix C

Bill of Materials

Presented here are the materials costs for the two-by-2two proof of concept system developed for this project, comprising 4 total machines.

Part No.	Vendor	Description	Price	Qty	Cost
VIA ML5000	`directron.com`	Mini-ITX x86 SBC	\$171.99	4	\$687.96
Casetronic Black ITX-2699R	`directron.com`	Mini-ITX Computer Case	\$62.99	4	\$251.96
Linksys WMP55AG	`newegg.com`	PCI Wireless A+G Network Adapter	\$79.99	4	\$319.96
Crystalfontz 633	`crystalfontz.com`	RS-232 LCD unit	\$55.00	4	\$220.00
Linx RXM-433-LC-S	`digikey.com`	433MHz Wireless Receiver Module	\$13.76	2	\$27.52
Linx TXM-433-LC	`digikey.com`	433MHz Wireless Transmitter Module	\$6.90	2	\$13.80
Linx ANT-433-SP	`digikey.com`	433MHz "SPLATCH" Planar Antenna	\$2.08	2	\$4.16
CUI RCJ-031	`digikey.com`	RCA jack, panel mount, black	\$0.71	2	\$1.42
CUI RCP-011	`digikey.com`	RCA plug, black	\$0.88	2	\$1.76
LED-PhotoTX1	`digikey.com`	Infrared LED	\$0.50	2	\$1.00
276-640	Radio Shack	Infrared Receiver Modules	\$3.69	2	\$7.38
	WPI ECE Shop	100Ω 5% resistor	\$0.10	2	\$0.10
	WPI ECE Shop	200Ω 5% resistor	\$0.10	2	\$0.10
	WPI ECE Shop	430Ω 5% resistor	\$0.10	2	\$0.10
	WPI ECE Shop	1KΩ 5% resistor	\$0.10	2	\$0.10
	WPI ECE Shop	10KΩ 5% resistor	\$0.10	4	\$0.20
	WPI ECE Shop	150KΩ 5% resistor	\$0.10	2	\$0.10
	WPI ECE Shop	100pF capacitor	\$0.40	2	\$0.40
	WPI ECE Shop	Miscellaneous wire and heat shrink tubing			\$1.00
				Total:	\$1540.12

Appendix D

Software Revisions and Driver Issues

This appendix lists the software and driver revisions used in this project. It also describes some of the issues encountered when using different third-party drivers and software.

D.1 Software and Driver Version List

- Debian 3.1
- Linux Kernel: 2.6.8-2-386
- Madwifi Atheros Chipset Driver: r1452
- Advanced Linux Sound Architecture (ALSA): 1.0.8
- Wireless Tools for Linux: 27-2
- TCPDump: 3.8.3

D.2 Other Software Evaluated During This Project (not used in final implementation)

- Linux Kernel 2.4
- Open Sound System (OSS)
- VideoLAN Client (VLC)
- Ndiswrapper 1.7
- Linuxant Driverloader
- Intel PRO/Wireless 2100 Driver for Linux

D.3 Software and Driver Issues

Several Linux kernel revisions were evaluated during this project. Recent stable revisions of the 2.6 Linux kernel proved to work the best for this project's application due to integrated ALSA support and compatibility with the latest wifi drivers.

Both ALSA and OSS were tested for audio support. Problems were encountered when using OSS for audio capture and playback. Moving to a Linux 2.6 kernel with integrated ALSA and installing the latest ALSA development packages (using apt-get) resolved these audio issues. The program alsamixer was used to adjust volume levels and capture from line-in.

Numerous wireless driver issues were encountered over the course of this project due to the continual development of wireless drivers on the Linux platform. Both native Linux driver and non-native Windows driver wrappers were tested for the wireless cards used during the project. Early laptop development used the Intel PRO/Wireless 2100 card. Both the native driver for Linux and the Windows driver using the Linuxant Driverloader were tested. After encountering issues with both drivers, they were eventually both able to work on an ad-hoc wifi network with multicasting.

For the Linksys WMP45AG card based on the Atheros chipset used in the final system, both native and non-native drivers were evaluated. The Madwifi native Linux driver initially failed to work in ad-hoc mode. The Ndiswrapper with the latest Linksys WMP45AG driver for Windows was tested and was able to create an ad-hoc network. However, this driver had issues with multicast streams and was generally unreliable. Finally, a later revision of the Madwifi native Linux driver was tested and was able to reliably connect to an ad-hoc network. Multicast transmission rates, however, were limited until the proper configuration was determined (see the configuration scripts madwifi-netsetup-a.sh and madwifi-netsetup-g.sh in the systems' root directory). As the Madwifi driver is in a continual state of development, new driver revisions should be evaluated as they are released.

Appendix E

Setup Procedure

This appendix details the procedure for setting up the two-by-two proof of concept system implemented in this project.

Step 1: Boot the machine with the 2.6 Linux kernel.

Step 2: Login as root with password "`m2mBrovn13`".

Step 3: From the root directory (`/`), run the shell script to setup the wireless network. For an 802.11a network run: "`sh madwifi-netsetup-a.sh`". For an 802.11g network run: "`sh madwifi-netsetup-g.sh`".

Step 4: Navigate to the subversion repository (e.g. `/home/m2m/project/orpheus/trunk/`).

Step 5: Use the make file to build the executables. (See the Makefile for options).

Step 6: Run the Control and Status Handler Application by invoking "`./control`" at the command line. You should now be able to control the system from the front panel UI.

Step 7: Run "`alsamixer`" to adjust volume levels and to capture the audio source at line-in (press F4).

Perform these steps on each of the machines.

Bibliography

[1] RF Link. 5.8GHz Wireless A/V Sender with Build-in I/R Remote Extender, Retrieved October 20 2005. http://www.rflinkusa.com/products_AVS5811.html.

[2] Apple Store. AirPort Express Base Station with AirTunes, Retrieved October 11 2004. http://store.apple.com/1-800-MY-APPLE/WebObjects/AppleStore.woa/72402/wo/Vy2UJiGQjNxH2Seyyw524L8MzIH/3.SLID?mco=7D88DA55&nplm=M9470LL%2FA.

[3] SMC Networks. EZ-Stream 11 Mbps Wireless Audio Adapter, Retrieved October 11 2005. http://www.smc.com/files/AQ/WAAB_ds.pdf.

[4] Apple Computer Inc. AirPort Express Technology Overview, 2004. http://switch.atdmt.com/action/apple_airportexpress_tech_overview.

[5] Bose Inc. Bose Link AL8 Homewide Wireless Audio Link. http://www.bose.com/controller?event=VIEW_PRODUCT_PAGE_EVENT&product=al8_wireless_transmitter_index_single.

[6] Federal Communications Commission. Code of federal regulations title 45 part 15, June 13 2005.

[7] IEEE 802.11. http://en.wikipedia.org/wiki/IEEE_802.11.

[8] Carrier Sense Multiple Access. http://en.wikipedia.org/wiki/Carrier_Sense_Multiple_Access.

[9] Intersil. A Condensed Review of Spread Spectrum Techniques for ISM Band Systems. application note 9820, May 2000. http://www.wingshing.com/product/intersil/AN9820.pdf.

[10] OSI Model. http://en.wikipedia.org/wiki/OSI_model.

[11] Transmission Control Protocol. http://en.wikipedia.org/wiki/Transmission_Control_Protocol.

[12] User Datagram Protocol. http://en.wikipedia.org/wiki/User_Datagram_Protocol.

[13] Real-time Transport Protocol. http://en.wikipedia.org/wiki/Real-time_Transport_Protocol.

[14] VIA. VIA Mainboards. http://www.via.com.tw/en/products/mainboards/mini_itx/epia_ml/index.jsp.

[15] Linksys. Dual-Band Wireless A+G PCI Adapter.
http://www1.linksys.com/products/product.asp?prid=526&scid=36.

[16] Crystalfontz Liquid Crystal Displays. CFA-633 Serial LCD.
http://www.crystalfontz.com/products/633/index.html.