A Wide Area Online Music Collaboration Emulation Platform

by

Bipin Pillay
B.Sc., Carleton University, 1998

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

A Wide Area Online Music Collaboration Emulation Platform

by

Bipin Pillay
B.Sc., Carleton University, 1998

Abstract

Over the last decade the internet has evolved at a tremendous rate, creating new technological and business opportunities. One such opportunity is real-time web-based collaborative music. Even though the internet has many advantages, one big disadvantage is the delay that is inherently present and that its behaviour cannot be predicted. The quality of a musician's experience is negatively affected by network delay. Therefore, the purpose of this thesis is to quantify through opinion scores of musicians' tolerances to an online jamming experience with various network delays.

A low cost network emulator was developed using Linux and freeware utilities to enable inserting various delays. Subjective feedback through a questionnaire was obtained from the musicians, and also quantitative data was captured and analyzed.

Our findings reveal that it is easier to play music than to just have a clapping rhythm between two musicians. Network delays of up to 50 ms could be tolerated for music sessions versus only 30 ms for clapping sessions.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AES	Audio Engineering Society
BPM	Beats Per Minute
FIFO	First in First Out
GUI	Graphical User Interface
HTB	Hierarchical Token Bucket
MAS	Mutual Anticipated Session
MIDI	Musical Instrument Digital Interface.
MPEG-4	Moving Picture Experts Group 4
NetEm	Network Emulator
NistNet	National Institute of Standards and Technology Network
NMP	Network Musical Performance
RMCP	Remote Music Control Protocol
RTP	Real Time Protocol
SIP	Session Initiation Protocol
TCN	Trace Control for Netem

Acknowledgments

Thanks to my supervisor, Dr. Thomas E. Darcie, for guidance, support, enthusiasm and motivation. I would like to thank Dr. Sudhakar Ganti for providing an incredible amount of support and interest during the course of this project. Thanks to Donna Shannon for taking care of all the administrative work and the friendship we have developed. Thanks to Kirk McNally for organizing the music sessions and setting up the equipment. Thanks also to my colleagues in the optics lab for the great friendships we developed over the years. I also thank my parents for their love and support over the years.

This thesis is dedicated to my father, R. K. Pillay.

Chapter 1

1.0 Introduction

1.1 Description

The emergence of web 2.0, broadly defined as web-based communities such as social-networking sites which facilitate sharing of ideas among web users, has generated a lot of attention to online communities. One such community, real-time web-based music collaboration (online jamming) is in its infancy. An online jamming application lets musicians from across the globe play together over the internet as if they were together at the studio. With online jamming, musicians can search the world for other musicians to create a band with no location bounds. However, even though the internet makes the world a smaller place, it does so with some disadvantages. The internet has some inherent delays, therefore real-time applications such as audio and video are affected. Online jamming is affected to a great extent, since the band is playing live and performing artists are highly sensitive to delay. Thus for this web application to be successful, the quality of the user's experience plays a crucial role. Therefore, the latency over the network has to be very low.

1.1.1 Internet Delays

The internet can be described as a network of networks and it can be broken down into a 3 tier hierarchy as shown in Figure 1.

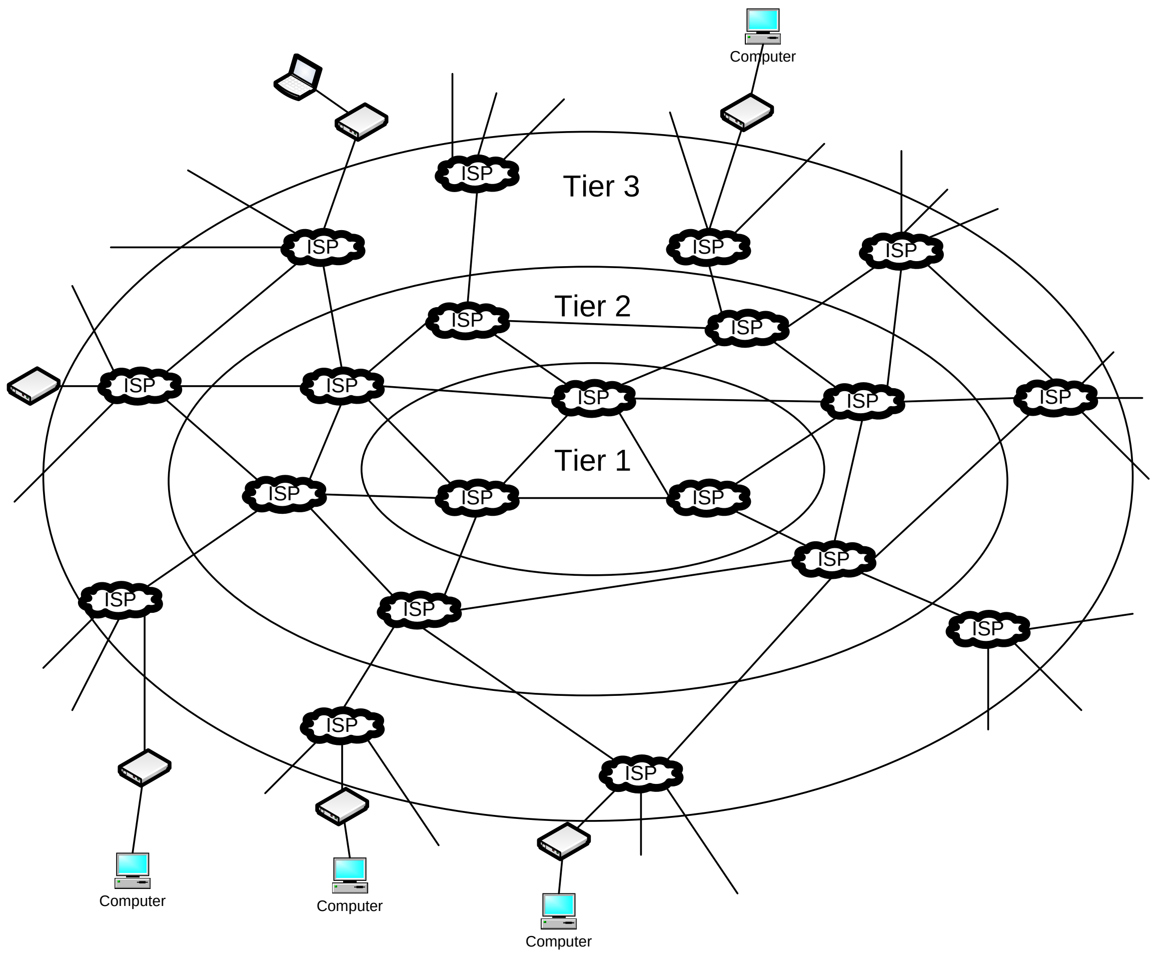


Figure 1 **Network Hierarchy**

The access networks at the edge of the internet are connected to the internet through a hierarchy of ISPs (Internet Service Providers). At the bottom of the hierarchy are the tier-3 ISPs or access ISPs which connect computers at home/office to tier-2 ISPs. Tier-2 ISPs have regional or national coverage and connect to a few tier-1 ISPs. Tier-1 is also known as the internet backbone and the Tier-1 ISPs are scattered across many geographical regions. In brief, the internet consists of thousands of tier-3 ISPs but only a few hundred tier-1 and tier-2 ISPs.

As a packet travels from its source to its destination, it will pass through some or all of the tiered networks and it will encounter several types of delays at each of the routers along the path. The contribution of each of these delay components can vary considerably.

The total delay is given by

Total Delay = Processing Delay + Queuing Delay + Transmission Delay + Propagation Delay

Processing delay is the time required to examine the packet's header to determine its next or final destination and can be in the order of microseconds. Queuing delay depends upon the length of the outgoing queue at the router and can be in the order of microseconds to milliseconds. A queue starts to build when the packet arrival rate is greater than the packet transmission rate. Transmission delay is the time required to push the entire packet onto the outgoing link and can be in the order of microseconds to milliseconds. Propagation delay is the time required for the packet to travel between the routers and is in the order of milliseconds. It is the distance between two routers divided by the propagation speed. The propagation speed depends upon the medium and is typically $2 * 10^8$ meters/second in optical fiber [23].

1.1.2 Why build an emulator?

With all the hype in online communities, a few companies have jumped into the online jamming space. These companies have created applications that can be installed on personal computers that enable the musicians to play together and record the music. Jitter is the variation in the packet arrival times that is caused by network congestion or route changes and the application can handle jitter by using jitter buffers. For these companies to test their software, they will require access to networks with delay and jitter characteristics representative of those encountered by targeted users. A more thorough test would be to run the jamming applications on a wide range of networks. Buying multiple networks can be expensive and also during the tests the companies do not have full control of the network. Therefore, a cost efficient solution is to use an emulator that can replicate any network using trace files or network models. This would also give them full control of the test environment.

Emulators available commercially are expensive, and cost on the order of thousands of dollars. To keep costs down, a preferred solution was to build a computer with Linux (Fedora) that can delay packets using an emulator called NetEm. The NetEm emulator is built into the Linux kernel. The Swiss Federal Institute of Technology (ETH) in Zurich has designed an emulator that can read a trace file that contains the ping times for a particular server or ISP location and NetEm will emulate the trace on a per packet basis. We use their application to perform the trace emulation.

In this thesis, we explain how the emulator was designed and used in a quantitative subjective test to measure musician's tolerance to delay. We analyze the tempo and ensemble difference between the players at various delays using Pro Tools, and analyze the subjective data collected from the musicians' reactions to each other's playing in the presence of delay.

1.2 Research Objectives

The main objectives of the research presented in this thesis are two-fold:

• Development of a portable Network Emulator

• To quantify tolerance for delay from musicians on their playback over the network through both subjective testing and quantitative data.

1.3 Thesis Contributions

This thesis provides three main contributions:

1.3.1 Network Emulator

A versatile Network Emulator that can be used to qualitatively assess musicians' tolerance to playing in the presence of network delay and that has broad applicability in future related studies.

The emulator features:

• A single Box with:

– Trace readable emulator

– Routing capability

– Accuracy of 1 ms

– Emulating multiple Network delay profiles

1.3.2 Subjective Testing Methodology

An original methodology for the subjective testing of musicians using the delay emulator, a commercial beta-release online collaboration application and Pro-Tools audio analysis software.

Methodology:

- Two musicians playing together, performing either a clapping session or a piano session with random delay
- Audio trace is analyzed for both tempo and ensemble consistency
- Musicians answered a subjective questionnaire about their experience after each session

1.3.3 Results

Live testing of twenty two musicians provides detailed quantitative understanding of tolerance to network delay in online collaboration. Results will have direct applicability to the design of the JamNow application, summarized as follows:

- Musicians can tolerate round trip delays of 50 ms for a musical piece
- Musicians can tolerate round trip delays of 30 ms for a clapping rhythm
- Round trip delays greater than 60 ms are detrimental to the experience

1.4 Thesis Outline

The thesis is organized as follows. In chapter 2, previous work in network latency tolerance is reviewed. In chapter 3, the implementation of the network emulator is introduced. Chapter 4 outlines both the quantitative results and the subjective results of the musicians and also compares the quantitative data with the subjective data. Finally chapter 5 presents the conclusion and future work.

Chapter 2

2.0 Background

Evaluating network performance is key to delay-sensitive applications such as VoIP, video conferencing and online music collaboration. The internet largely works on a best effort delivery basis which is acceptable for non-real-time applications such as web surfing, ftp and email. However, real-time audio and video is an unforgiving test of network performance [2].

Network delays can range from one millisecond to hundreds of milliseconds depending upon the physical distance of the connection and routing delays. The physical distance from the west coast to the east coast of the United States is approximately 3000 miles, thus at 70% of the speed of light, the shortest one-way delay is approximately 24 ms [1,2]. The delays would be larger than 24ms since no actual fiber path exists that traverses the US in 3000 miles, as fiber routes tend to follow rings and roads.

It is unclear to what degree and under what conditions (tempo, genre, practice, expectations) maximum tolerable delay can be increased. Therefore, to better understand acceptable conditions, we require a method to emulate various network conditions and a method for subjective assessment.

2.1 Emulators

An emulator allows developers and testers to observe real-time interactions between different hardware or software modules in a design, whether it is a network or an electronic device.

There are a few freeware emulators. They are available from Dummynet, NistNet and NetEm. We only considered NistNet and NetEm as they met our requirements of running on Linux, while Dummynet runs on FreeBSD. These emulators can introduce fixed-delay, variable-delay and bandwidth limitations but they cannot account for long-term internet traffic behaviour [9]. Internet traffic behaviour over the long-term is very erratic and these emulators cannot reproduce the long-term network dynamics. Therefore, the capability to replay a large packet trace that was previously generated or captured is invaluable. A trace file can be generated in Linux by using the ping command and sending the output to a file.

To overcome this limitation, the Swiss Federal Institute of Technology designed a trace reader called ReplTrc for NistNet [9] and TCN for NetEm [7,8]. This project implemented the TCN for NetEm for reasons discussed in chapter 3. By patching the trace reader package with the respective emulator, an emulator that can replicate long-term internet traffic behaviour was developed. The Swiss Federal Institute of Technology uses the trace reader for performance evaluation of distributed applications. We will be using the application specifically for online music jamming network testing. TCN is available for free at http://tcn.hypert.net/ .

There are also a few commercial emulators available from companies such as Anue Systems, Apposite Technologies, Packet Storm, Network Nightmare, Itheon and Simena. Most of these emulators are expensive, costing thousands to tens of thousands of dollars, and they do not have the capability to read in a ping trace file. They can capture and replay traffic but this does not meet our requirements. They can also function at wireline speeds of 10/100/1000 megabits per second. Our data rate requirements are significantly lower and the freeware NetEm emulator meets them with a large margin.

2.2 Online Jamming

2.2.1 Research

In 1998, an AES (Audio Engineering Society) white paper on networking audio brought light to online music research [22]. Since then the internet has evolved significantly and there has been a lot of research in this area. Topics investigated include better protocols, lower packet processing latency, and musicians' tolerance to delay. Some of this related research is discussed below.

In 1997, researchers at the Hebrew University of Jerusalem built a system that allowed musical performers to play together in multiple session groups. The players were playing on their MIDI controllers, using the general MIDI protocol. The network was built on the Transis architecture that supports efficient and reliable multicast sessions [16].

In September of 2000, researchers at McGill University had a jazz band perform live in Montreal while they had recording engineers mixing the 12 channels of uncompressed audio at the University of Southern California in Los Angeles. This was streamed live over the internet to the recording engineers but all the musicians were on the same stage in Montreal [6]. Since all the musicians were in the same room there were no ensemble issues, but this was the very first live performance over the internet.

Researchers at Stanford University studied musicians' reactions over various network delays ranging from 0 ms to 77 ms. They analyzed the rhythmic accuracy for a duo clapping session between the musicians. Their study revealed that delays longer than 11.5 ms caused the tempo to decelerate while delays shorter than 11.5 ms caused the tempo to accelerate. The optimal delay was found to be 11.5 ms [1].

In 2001, researchers at University of California Berkeley presented a case for Network Musical Performance (NMP) as a practical multimedia application. Their system combined MPEG-4 for sound synthesis, RTP and SIP for networking, and MIDI for musical Control. Their network only spanned from UC Berkeley to Stanford University and Caltech, with delays ranging from 6 ms to 33.5 ms. The nominal latency was readily apparent at approximately 30 ms causing depressed keys on the electronic piano not to sound and released keys sometimes keep sounding for a short time period. Their experiments used hosts that were connected directly to enterprise routers and therefore were able to use low latency routes. They concluded that last mile technologies dominate the end-to-end delay between two hosts and can result in a total latency too high for a useable NMP [24].

In 2004 and 2005, the Viterbi School of Engineering and the Thorton School of Music at the University of Southern California performed a music ensemble session between two award-winning pianists who were in the same room but the audio was delayed randomly between 0 ms and 100 ms. Therefore, they could see each other and their reactions but they could only hear each other through the delayed network via headphones. They showed that delays of up to 50 ms could be tolerated and if they practiced more on the network, then they could tolerate up to 65 ms delays [20, 21].

In 2004, Nicolas Bouillot came up with the “conductor synchronization scheme” approach. This approach lets the musician hear two streams, one from themself and the other from the conductor. The conductor’s broadcast could be a click track with a time stamp of the server. They concluded that if the musicians played synchronously with the conductor stream, then the mix will be synchronous. This architecture does require that the music have many interactions between musicians such as a symphony.

Researchers at Pompeu Fabra University and Portuguese Catholic University showed that there is an inverse relationship between network delay tolerance and tempo. They show that more network delay can be tolerated for slower tempos [5]. This also reveals that depending upon the tempo chosen for the sessions the results can vary between research labs.

Researchers at Ibaraki University proposed a new protocol called Mutual Anticipated Session (M.A.S) to compensate for network delays. One player’s musical performance precedes the other player’s musical performance. Using this approach, the researchers were capable of achieving satisfactory levels of comfort from the players.

Barbosa in his thesis performs a survey of all computer-supported collaborative music applications [18].

2.2.2 Commercial

The two leading companies providing live online jamming experiences are JamNow and eJamming; their respective websites are www.jamnow.com and www.ejamming.com.

JamNow based its technology on a client-server model and plans to minimize the delay from all sources by using low-delay audio compression and network transmission.

eJamming attempts to minimize the latency using three approaches. First, the application uses audio compression to reduce the file size before being sent on the network. Second, it uses a peer-to-peer configuration in which they claim the musicians are connected directly to each other. Third, the application time stamps the packets to synchronize the music [25]. In reality, these methods increase the latency within the application by increasing the processing time to synchronize, compress and decompress the data. Also, the peer-to-peer connections do not, in most instances involving traffic routed between different network service providers, reduce delay.

Another popular website is NINJAM, written by Cockos Incorporated and Brennan Underwood. It uses the client-server model and sends compressed audio [18]. It uses the RMCP (Remote Music Control Protocol) designed by researchers at Waseda University to increase the delay to a musical relevant quantity with very small deviation [11,19]. Effectively, this means synchronizing to other artists audio that was generated a measure previously makes it particularly challenging to perform naturally, as transitions must be anticipated.

Chapter 3

3.0 Implementation

3.1 Design of Music Network Emulator

The objective was to build a “box” at minimal cost that can emulate delays in a network and route packets between the music server and the clients. Linux was chosen as the operating system and freeware emulator packages were installed to keep costs down. The online jamming software that was employed on the clients and the server was the beta version of a popular online jamming website http://www.jamnow.com/ which was provided by Lightspeed Audio Labs.

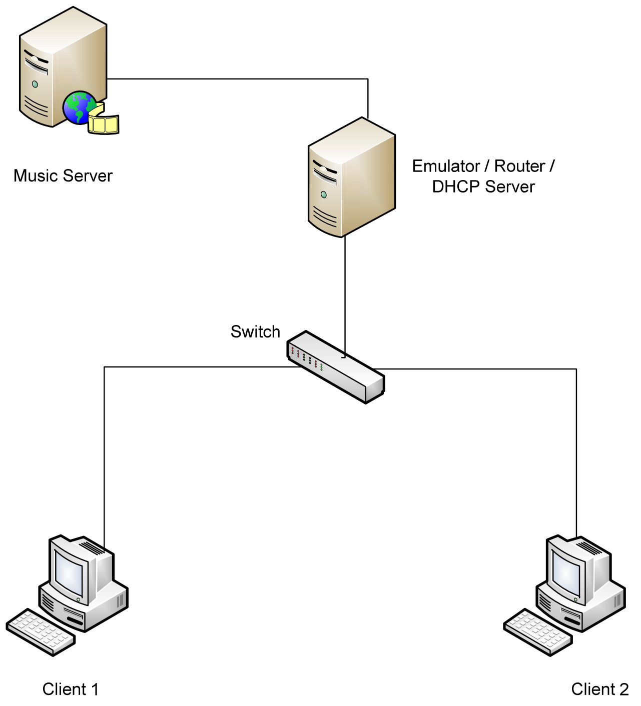


Figure 2 **Music Network Setup**

The Music Network consists of a music server, an emulator and two clients, as shown in Figure 2. The Linux emulator was built with Fedora Core 5 and kernel version 2.6.18.

Since we only had two ports on the emulator for the music server and the two clients, a multi-port GigE ethernet switch was used for the clients. To provide the IP addresses to each of the clients, a DHCP server was installed on the emulator. The ethernet port on the emulator that is connected to the music server is set to a static address since the music server has a static address.

We could not use an external router in place of a switch because then the clients could not establish a connection to the music server. The external router can only see the emulator and cannot translate the music server's address over to the clients. Therefore any requests from the clients to the music server are rejected by the external router. Also, having an external router adds unnecessary complexity to the network because then there are two routers in the traffic path.

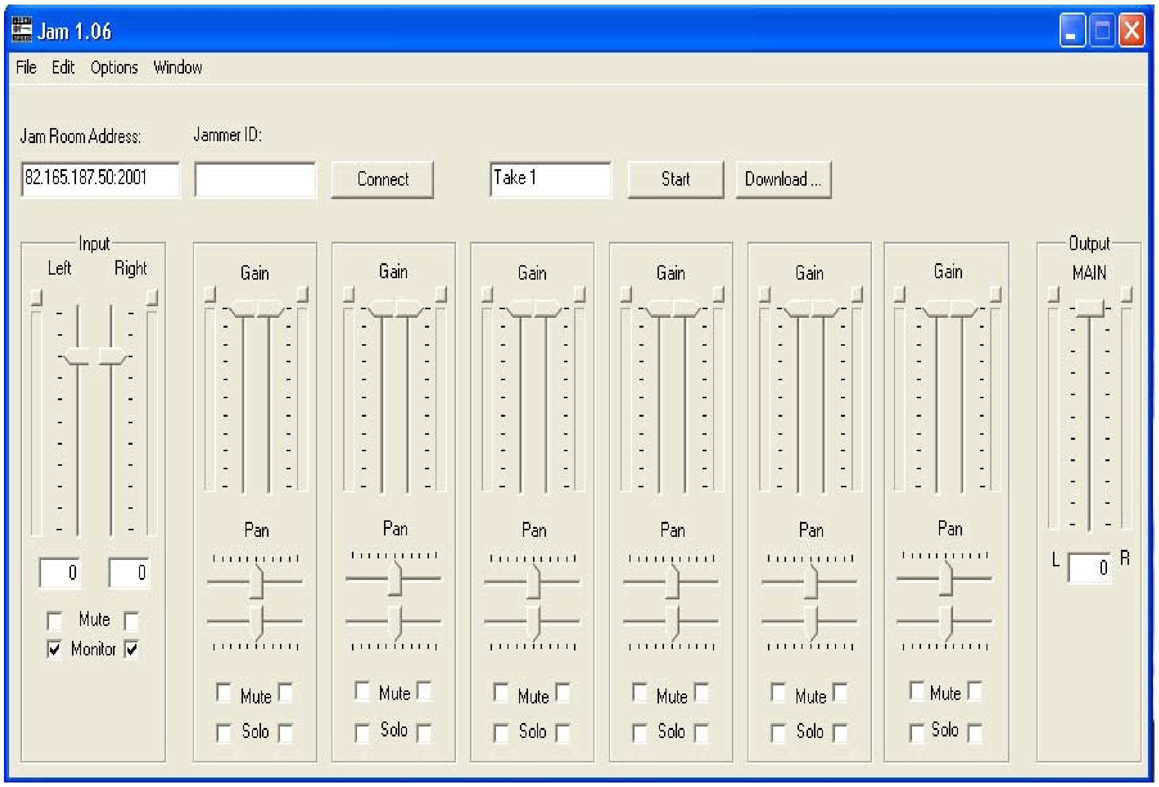
3.1.1 Online Jamming Application

Jam, the online jamming application, was provided by Lightspeed Audio Labs. The application uses the client server model therefore the application runs on both the server and the clients.

Client Graphical User Interface

The main panel of the Jam client graphical user interface (GUI) is shown in Figure 3. The controls are:

- General – These are along the top
- Input Panel – This is the leftmost group of controls
- Performer Slots – There are six group of controls for six musicians
- Output Panel – This is the rightmost group of controls



Jam Room Address: 82.165.187.50:2001

Jammer ID:

Take 1

0 0 0

☐ Mute ☐

☑ Monitor ☑

Figure 3 Jam – Online Jamming Application User Graphical Interface

3.1.1.1.1 General Controls

The general controls consist of:

Jam Room Address - IP address with port number of the server to connect to
Jammer ID - User ID
Connect button - Connect to the server
Start button - Start Recording on server
Download button - Download the recorded session as a wave file to your computer for listening and/or reviewing

3.1.1.1.2 Input Panel

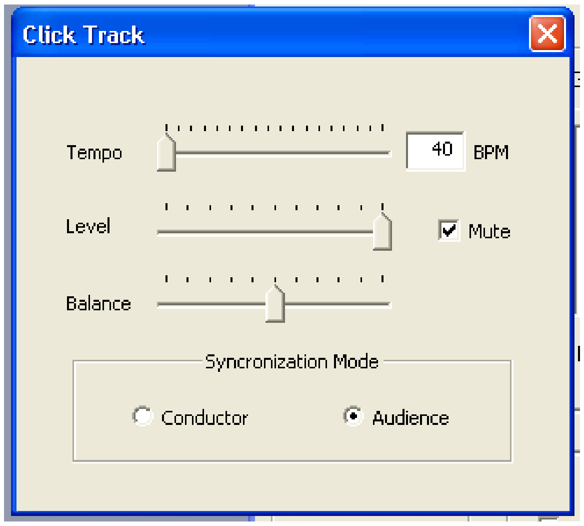
The input meter shows local left and right input channel levels as sent to the mixer. The meter readout is in decibels and it shows the signal levels in three different colors: red, yellow and green.

3.1.1.1.3 Performer Slots

The input meter shows local left and right input channel levels after the mix gain is imposed. Each client can independently adjust the levels of all others. The meter readout is in decibels and it shows signal levels in three different colors: red, yellow and green.

3.1.1.1.4 Click Track Window

The Click Track window is shown in Figure 4.



40 BPM

[✓] Mute

Conductor [✓ Audience]

Figure 4 Click Track Window

The click track provides an auditory (click) cue to aid a performer keeping the beat. The window permits the performer to set the following controls:

- Numerical readout of beats per minute (BPM), which can be set by the slider or typed in
- Level Slider sets the level of click
- Mute the click by clicking the soft button
- Balance is left/right balance of click
- Synchronization mode: Conductor or Audience

The conductor mode gives the perception that both the clients are hearing the click track at exactly the same time, as if they were seeing the conductor and therefore compensates for delay. While the audience mode does not compensate for delay, as if the clicks were separate events.

Online Jamming Application Delay Budget

An overall application delay budget is included in Table 1 [28]. Time measurements are given in microseconds (µs) or milliseconds (ms).

Location	Module	Delay	Comment
Client	Audio I/O	<1 ms	Headphones
Client	Audio Driver	<6 ms	
Client	Audio Encoder	<50 µs	
Client	Receive FIFO	5 ms	Jitter Buffer
Client	Audio receive thread timing and decode	3 ms	
Server	Input FIFO	5 ms	Jitter Buffer
Server	Audio decode, error mitigation, mix, audio encode	<100 µs	
TOTAL		**~20 ms**	

Table 1 Jam – Online Jamming Application Delay Budget

The JamNow application has an upper bound application delay of 20 ms. The results did not take into account the 20 ms application delay because the delay variance is unspecified and the delay varies between applications. The application handles packet loss by error concealment.

3.1.2 Choice of Emulator

The online jamming application from Lightspeed Audio Labs has a maximum packet rate of 195 packets/second (1 packet every 5 ms) with 128 bytes per packet. This gives us a raw data rate of approximately 200 kilobits/second or a minimum requirement of 5 ms resolution for the emulator timer. Since this data rate is quite low, we can use either of the popular Linux emulators:

i. Nistnet
ii. NetEm

NIST Net is a Linux kernel module that intercepts network packets and performs traffic control [14].

NIST Net can inject three types of delay:

1. Fixed delay
2. Variable delay
3. Mathematical distribution delay

NIST Net is capable of delaying packets, generating packet loss, and duplicating packets. The timer in Nisnet has a high resolution of 121 microseconds because it is installed with a Real Time Module that runs at 8192 Hz [9].

NetEm is part of the linux kernel versions 2.6 and higher. It also intercepts packets and performs various forms of traffic control. In addition to generating delay, duplication and packet loss, NetEm has the capability to corrupt packets. It also has fixed delay, variable delay and mathematical distribution delay. The timer in NetEm has a resolution of 1 ms because of the limitation of the linux timer resolution. Since NetEm is built into the kernel and will be available with new kernel releases, we decided to use NetEm.

Our requirement was also to have the emulator read trace files to emulate specific network connections. TCN (Trace Control for NetEm) developed by the Swiss Federal

Institute of Technology Zurich was used as an extension to NetEm. TCN delays, drops, duplicates and corrupts packets based on values in a trace [8].

This package has three main features:
1. Monitoring Traffic
2. Traffic Classification
3. Traffic Manipulation

TCN can handle a maximum packet rate of 80,000 packets/second with 54 bytes per packet. This gives us a raw data rate of 34.56 Megabits/second which is significantly higher than our requirement.

3.1.3 Downloading and Installing TCN

The main website for TCN can be found at http://tcn.hypert.net. This page has full instructions on how to install TCN on linux. It also has links to a kernel patch, IProute2 patch and documentation.

At the time of this writing, the most current version of TCN is version 1.0. You must use Fedora linux kernel version 2.6.18 or higher. TCN does not work with older versions of the kernel, even though NetEm is part of linux kernel 2.6.7 and higher.

The steps to install TCN are: [7]

- i. Source code of tc (traffic control) and source code of NetEm kernel module have to be patched.
- ii. The frequency of the linux kernel’s internal timer has to be changed to 1000 Hz (1 ms) from the default value of 250 Hz (4 ms)

3.1.4 Configuring the Queuing Disciplines

Linux Traffic Control consists of the following components:

1. Queuing Disciplines
2. Classes within a Queuing Discipline
3. Filters

A queuing discipline represents the rules to insert and remove packets from a queue. Each network connection has an associated queuing discipline that controls how enqueued packets are treated on that device [10]. NetEm is a queuing discipline (qdisc) that has to be configured with tc, the traffic control tool of linux [7].

Filters can be used to distinguish between classes of packets within a queuing discipline and each of the classes can be processed in a specific way. A traditional queuing discipline is shown in Figure 5.

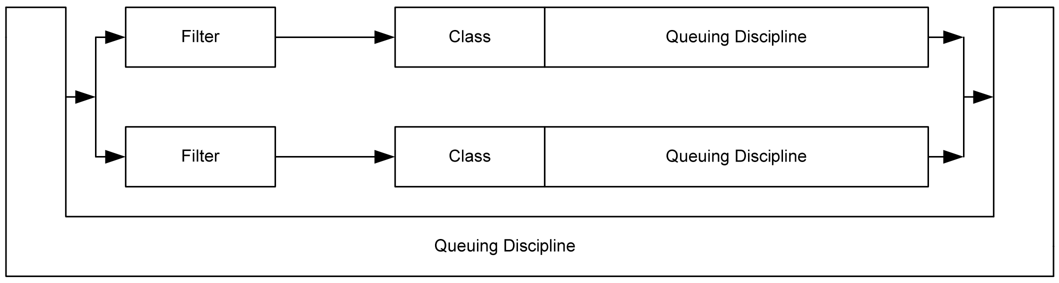


Figure 5 Traditional Queuing Discipline

A traditional queuing discipline consists of a root queuing discipline, which contains a filter with its own class ID and queuing discipline [26].

There are 11 types of queuing disciplines that are currently supported in Linux:
1. Class Based Queue (CBQ)
2. Hierarchical Token Bucket (HTB)
3. Clark-Shenker-Zhang (CSZ)
4. First In First Out (FIFO)
5. Priority
6. Traffic Equalizer (TEQL)
7. Stochastic Fair Queuing (SFQ)
8. Asynchronous Transfer Mode (ATM)
9. Random Early Detection (RED)
10. Generalized RED (GRED)

11. Diff-Serv Marker (DS_MARK)

Hierarchical Token Bucket (HTB) was implemented because it is accurate, simpler to configure and very flexible at distributing bandwidth among several classes [3]. HTB ensures that the amount of service provided to each class is at least the minimum of the amount it requests and the amount assigned to it. [15]

Two HTB queuing disciplines have been added, one for each of the internet interfaces on the emulator. The queuing disciplines on the ethernet interface eth0 controls the data traffic from the clients to the server, while the queuing disciplines on the ethernet interface eth1 controls the data traffic from the server to the clients.

The implementation of the queuing discipline is illustrated in Figure 6.

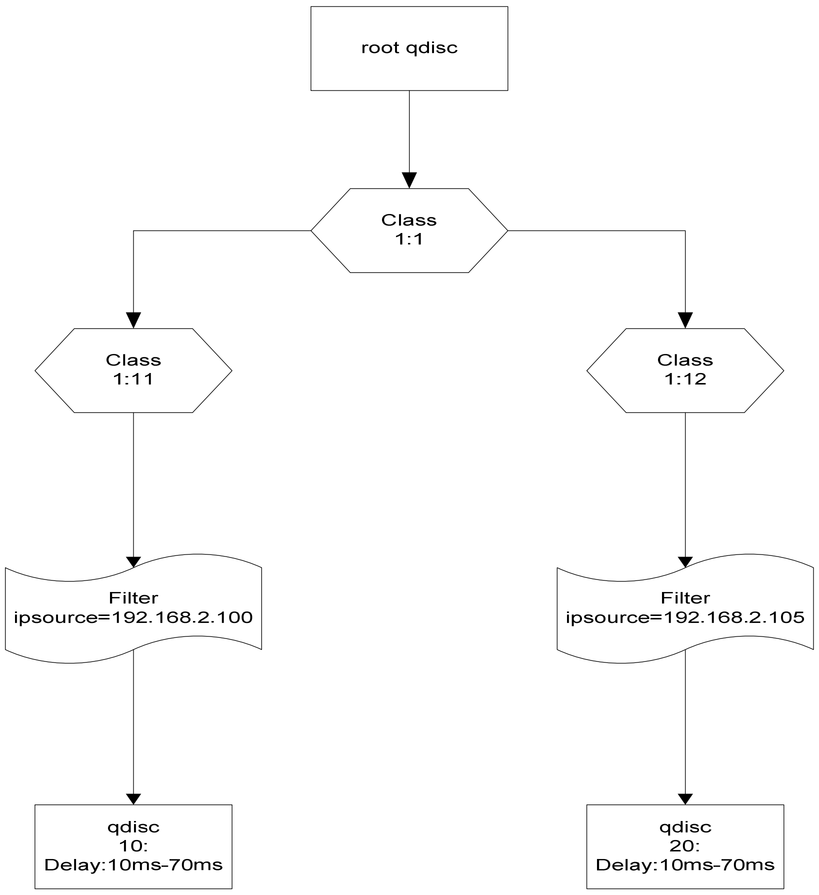

root qdisc
Class 1:1
Class 1:11
Class 1:12
Filter ipsource=192.168.2.100
Filter ipsource=192.168.2.105
qdisc 10: Delay:10ms-70ms
qdisc 20: Delay:10ms-70ms

Figure 6 Queuing Discipline Setup

First, a root class [class 1:1] is created. From the root class, two subclasses are created:

i. Class 1:11 will control traffic from/to the server and client 1

ii. Class 1:12 will control traffic from/to the server and client 2

The HTB queuing discipline was configured as shown below:

The command is shown and then a brief description is given.

command:

```
tc qdisc add dev eth0 handle 1:  root htb
tc qdisc add dev eth1 handle 1:  root htb
```

These commands create the root HTB queuing discipline on devices eth0 and eth1.

command:

```
tc class add dev eth0 parent 1: classid 1:1 htb rate 100mbps ceil 100mbps
tc class add dev eth1 parent 1: classid 1:1 htb rate 100mbps ceil 100mbps
```

This creates a root class with a maximum bandwidth of 100 Mbps. To avoid each client from borrowing bandwidth from the other, the ceil parameter was set to the rate parameter. The rate parameter guarantees the minimum bandwidth to the client and the ceil parameter sets the maximum bandwidth if other clients are over-provisioned. Therefore, bandwidth borrowing is allowed if the ceil parameter is greater than the rate parameter.

command:

```
tc class add dev eth0 parent 1:1 classid 1:11 htb rate 100mbps
tc class add dev eth0 parent 1:1 classid 1:12 htb rate 100mbps
tc class add dev eth1 parent 1:1 classid 1:11 htb rate 100mbps
tc class add dev eth1 parent 1:1 classid 1:12 htb rate 100mbps
```

Here, classes 1:11 and 1:12 are created to control flows to/from client 1 and client 2 respectively.

The next step is to configure the filters that place the packets in each class, as shown in Figure 7.

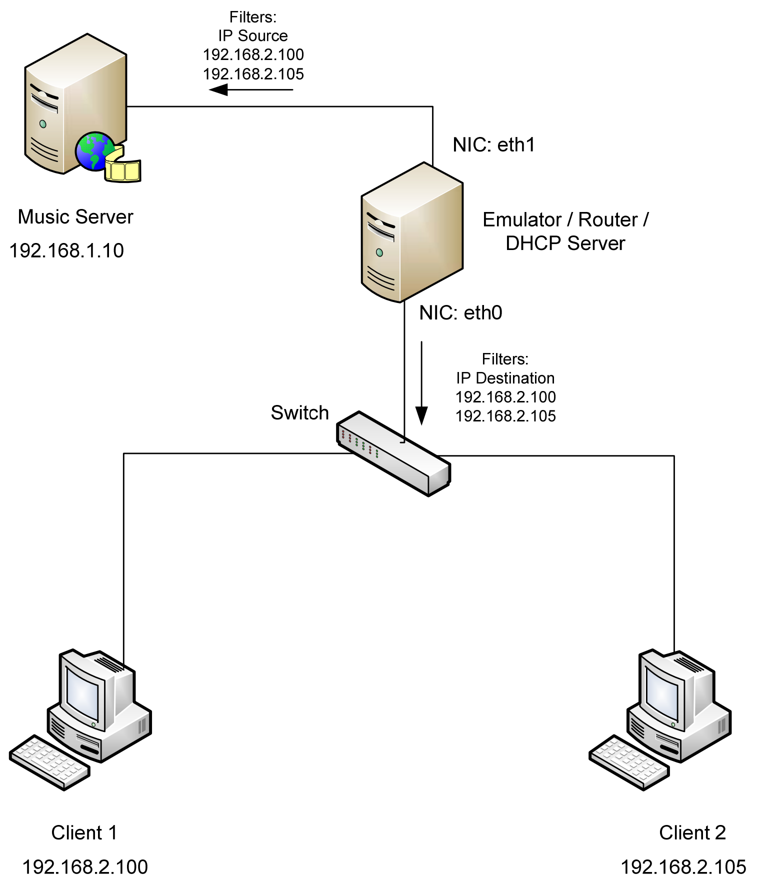


Figure 7 tc Filter Setup

command:

```
tc filter add dev eth0 parent 1: protocol ip prio 1 u32 match ip dst 192.168.2.100 flowid 1:11
tc filter add dev eth0 parent 1: protocol ip prio 1 u32 match ip dst 192.168.2.105 flowid 1:12

tc filter add dev eth1 parent 1: protocol ip prio 1 u32 match ip src 192.168.2.100 flowid 1:11
tc filter add dev eth1 parent 1: protocol ip prio 1 u32 match ip src 192.168.2.105 flowid 1:12
```

The filters match to the destination IP addresses 192.168.1.100 (client 1) and 192.168.1.105 (client 2) for network interface card eth0; and to the source IP addresses 192.168.1.100 (client 1) and 192.168.1.105 (client 2) for network interface card eth1. This is because NetEm processes packets on the egress of the network interface card. All packets heading to the music server will be processed by the qdisc on network interface card eth1, and all the packets heading to the clients will be processed by the qdisc on network interface card eth0. This allows us to insert different delays for upstream and downstream flows, if required.

Figures 8 and 9 illustrate the queuing discipline that was implemented on each of the network interface cards.

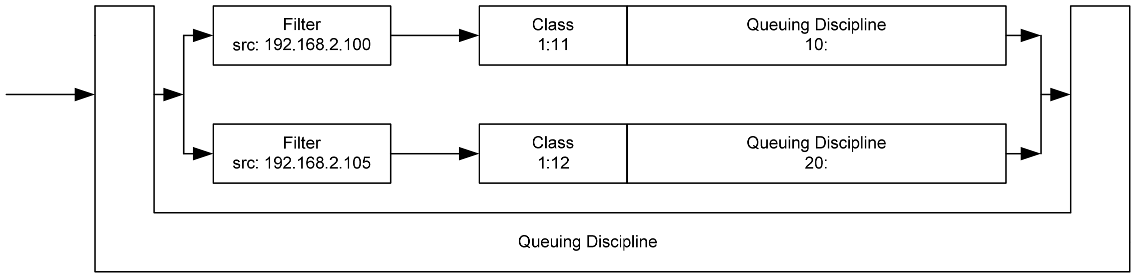


Figure 8 eth1 Queuing Discipline

Figure 8 illustrates that when a packet enters network card eth1; the classifier in the main queuing discipline immediately handles it by using the filter to determine if its source address is 192.168.2.100 or 192.168.2.105 to select the class the packet belongs to. Once the packet is assigned to a class, the corresponding queuing discipline delays the packet as set by the user.

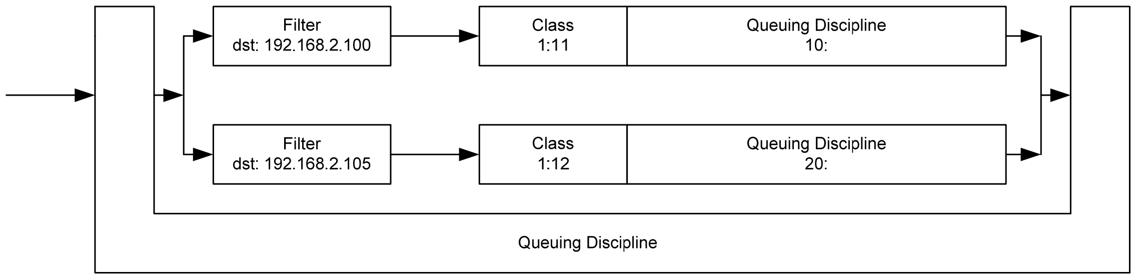


Figure 9 eth0 Queuing Discipline

Figure 9 illustrates that when a packet enters network card eth0; the classifier in the main queuing discipline immediately handles it by using the filter to determine if its destination address is 192.168.2.100 or 192.168.2.105 to select the class the packet belongs to. Once the packet is assigned to a class, the corresponding queuing discipline delays the packet as set by the user.

3.1.5 Configuring the Emulator as a Router

Both the network interface cards were set to static addresses. The network interface card eth0 is set as 192.168.2.0 for the 2.0 sub-network (client sub-network) and the network interface card eth1 is set as 192.168.1.0 for the 1.0 sub-network (music server sub-network).

For the emulator to route packets to the appropriate clients and the music server, static routes were setup on the emulator.

commands:

```
route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.1 dev eth1
route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.2.1 dev eth0
```

The route command specifies the destination network, the ethernet device and address the data is to be sent on.

Since IP forwarding is disabled by default, it was enabled to allow packets to traverse between the network interface cards. Masquerading was also enabled to perform network address translation on network interface card eth0. The masquerading setup is illustrated in Figure 10.

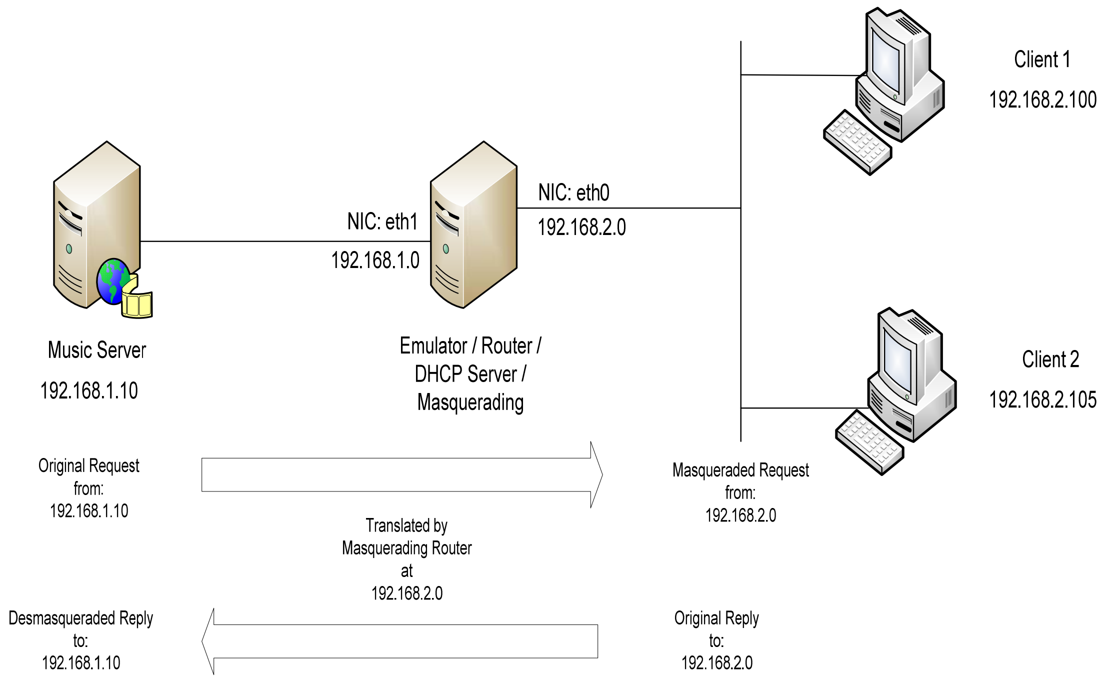

Client 1
192.168.2.100
NIC: eth1
192.168.1.0
NIC: eth0
192.168.2.0
Music Server
192.168.1.10
Emulator / Router / DHCP Server / Masquerading
Client 2
192.168.2.105
Original Request from: 192.168.1.10
Masqueraded Request from: 192.168.2.0
Translated by Masquerading Router at 192.168.2.0
Desmasqueraded Reply to: 192.168.1.10
Original Reply to: 192.168.2.0

Figure 10 **Masquerade Configuration**

The emulator substitutes its own IP address (192.168.2.0) for the music server's datagrams and transmits the datagram to the clients. The clients believe that they received a datagram from the emulator and generate a reply. Upon receiving this datagram, the emulator (masquerading host) finds the association in its masquerade table and reserves the substitution it had performed and then transmits the datagram to the music server. [13]

3.1.6 Configuring the Emulator as a DHCP Server

A DHCP server was installed on the emulator on the network interface card eth0, because the clients were setup to dynamically accept an IP address. This gave us the flexibility to switch the computers between the private network and the university public network.

command:

```
range dynamic-bootp 192.168.2.100 192.168.2.105
```

The range of IP addresses the DHCP server will issue was set from 192.168.2.100 to 192.168.2.105 for the client machines.

command:

```
option routers 192.168.2.1
```

The default gateway was set as 192.168.2.1 (client sub-network).

Chapter 4

4.0 Subjective Testing Methodology

The following experiments illustrate a test of ensemble accuracy with network delays ranging from 10 ms to 70 ms. The results did not take into account the application delay because this varies between applications and newer revisions.

Eleven pairs of subjects were recorded for both duo-clapping and playing the piano. A simple inter-locking rhythm was chosen for the duo-clapping and “twinkle twinkle little star” was chosen for the piano ensemble.

Ensemble accuracy was measured by analyzing both the tempo of each player individually and also the ensemble time difference between the players. Subjective results were also collected and compared to the tempo and ensemble difference. The piano performance was also compared to the duo-clapping in terms of playing comfort and tempo.

4.1 Test Setup

4.1.1 Audio Setup

The audio setup is shown in Figure 11.

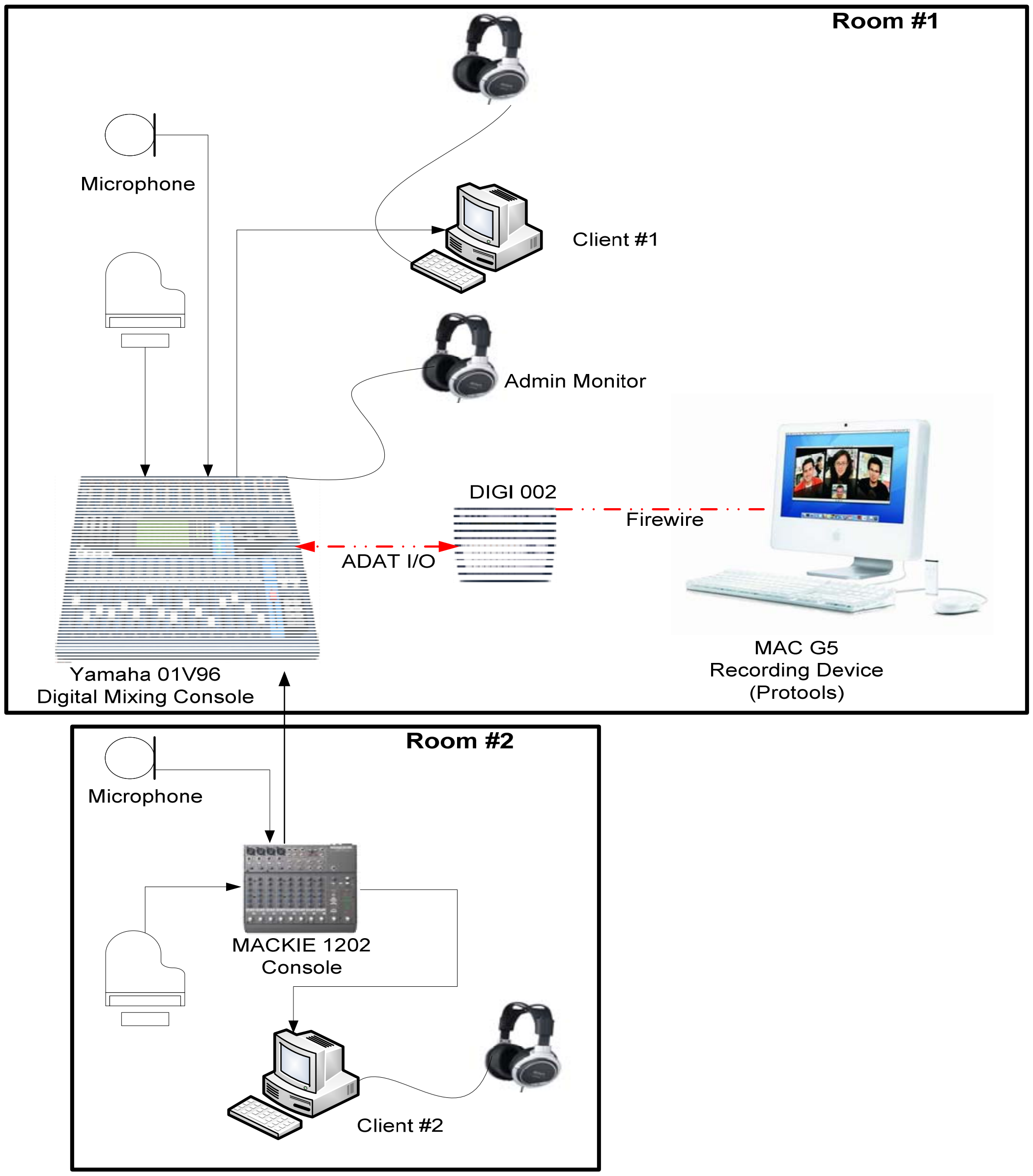


Figure 11 Audio Setup Topology

The audio setup was installed in two rooms that were acoustically and visibly separated from each other.

Room #1 was equipped with:

i. Client 1 (computer)

ii. Piano and microphone for the musician

iii. Yamaha Digital Mixing Console, Model #01v96

iv. A DIGI002 music production system and an Apple MAC G5 computer for recording the session

Room #2 was equipped with:

i. Client 2 (computer)
ii. Piano and microphone for the musician
iii. A Mackie Console, Model #1202 Mixer

The Mackie console mixer is used in room #2 only to avoid running long cables for the microphone and the piano back and forth between the two rooms. The Mackie console mixer's output is connected to the Yamaha 01v96 digital mixing console. The Yamaha 01v96 digital mixing console was used as the main mixing console for the two musicians. The mixed output was sent over ADAT to the DIGI002 which sent the audio over firewire to an Apple MAC G5 with Pro Tools LE recording software. The DIGI002 with the Pro Tools LE software is a firewire based music production system that enables us to record exactly how the musicians react to the network delays and click track.

The online jamming application, Jam, was running on each of the client computers and the synchronous mode for the click track was set as conductor. The conductor mode gives the perception that both the clients are hearing the click track at exactly the same time. There is no network delay compensation being done by the online jamming application on our recordings. The application does provide some kind of delay compensation on the recordings at the server to compensate for the network delay but we do not analyze the recordings on the music server.

4.1.2 Flow of Audio Traffic

The flow of audio traffic between the clients is depicted in Figures 12 and 13.

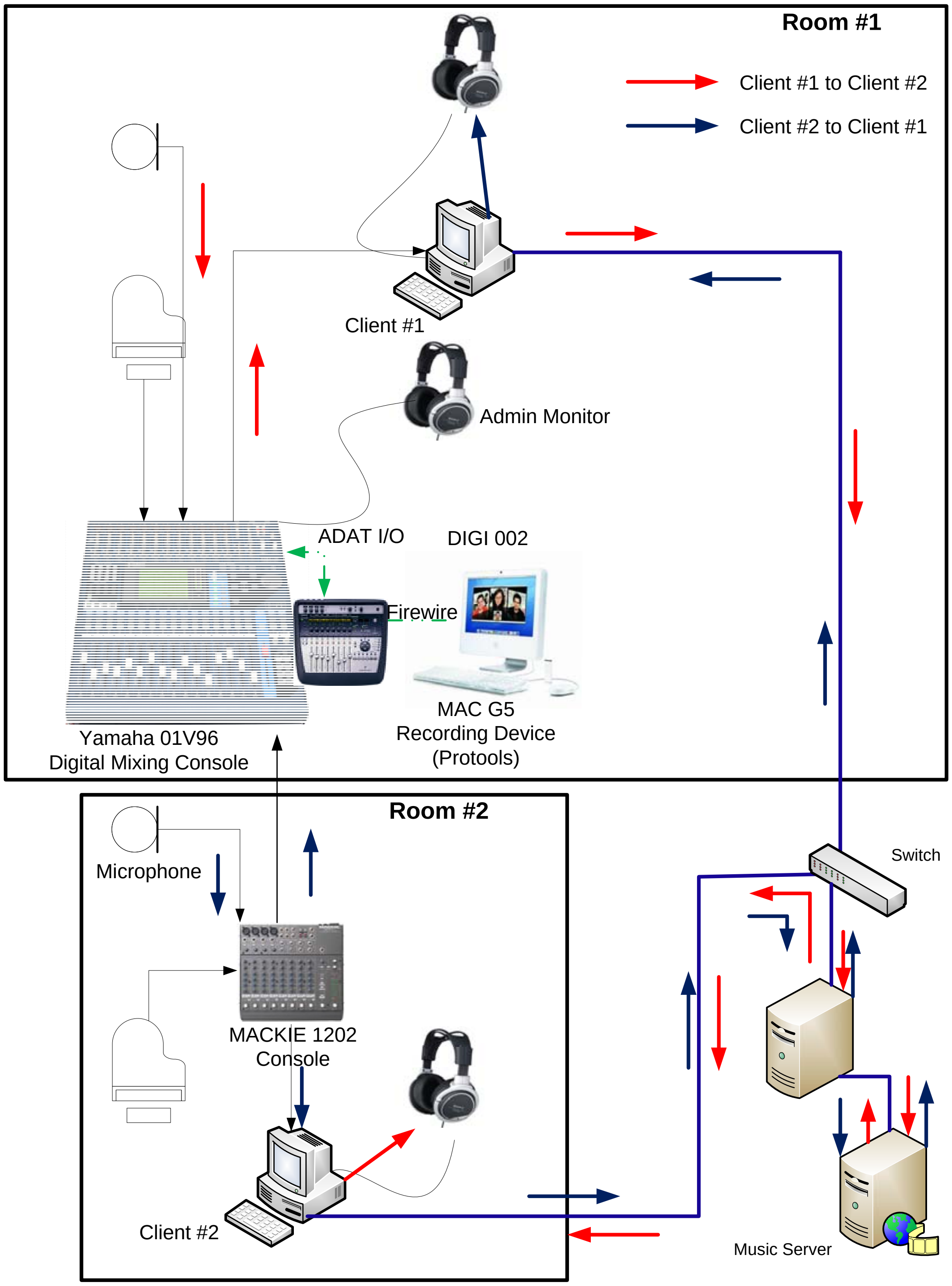


Figure 12 **Network Level Audio Traffic Flow**

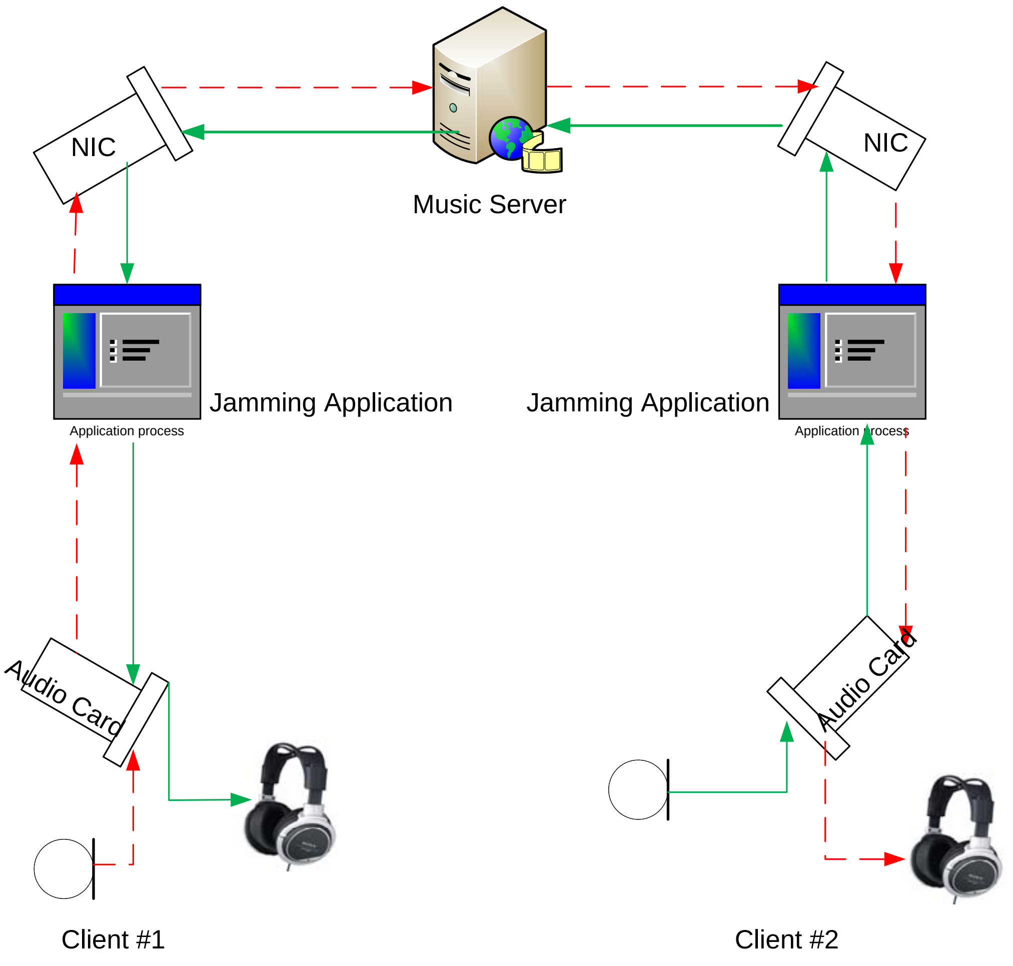


Figure 13 Low Level Audio Traffic Flow

When there is input on the microphone on client1, it is received on the audio card and then read by the jamming application. The jamming application converts this audio input into 128-byte packets and sends it over the Ethernet network to the music server. The server then decides which client/s should receive this packet; in this scenario it is sent to client2. It is received by client2's network interface card and then read by the jamming application. The jamming application outputs it to the audio card to be heard by the musician. This process is reversed for audio generated by client2.

4.2 Subjects and Tasks

The musicians were chosen randomly, with the only requirements being that they be able to play the piano, read music and play with a metronome.

They were given two separate tasks:

1. Clap at 90 beats per minute to the inter-locking rhythm shown in figure 14
2. Play "Twinkle Twinkle Little Star" shown in figure 15 on the piano in unison at 90 beats per minute

The clapping notation is shown in Figure 14 and the music notation is shown in Figure 15.

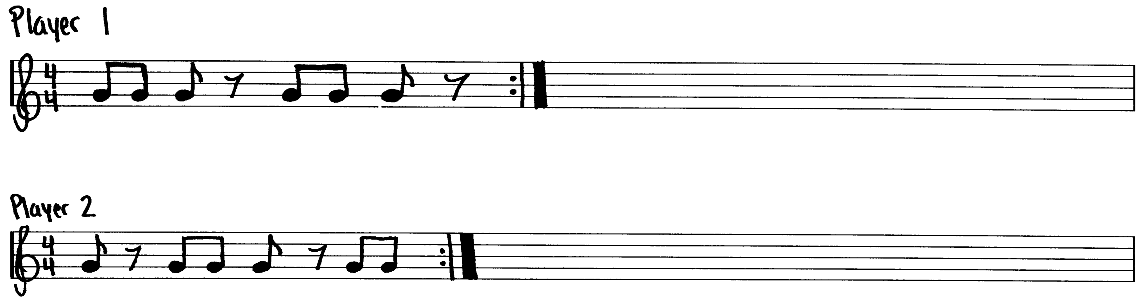


Figure 14 Clapping Content

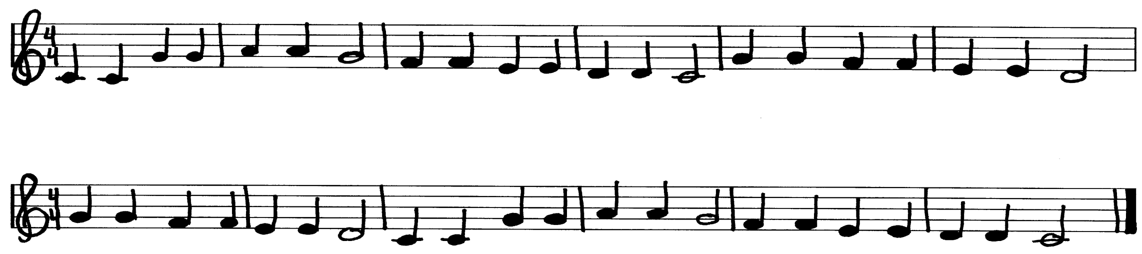


Figure 15 Piano Content

Each session had two musicians playing together, performing one of the two tasks. The session had seven trials with network delays between 10 ms to 70 ms (10 ms increments) in random order. Therefore, there were seven trials for clapping and another seven trials for the "twinkle twinkle" on the piano. For each trial to be approximately sixty seconds in length, the musicians were asked to clap to the inter-locking rhythm for at least sixty seconds, or to play "Twinkle Twinkle Little Star" twice, depending on the task. The musicians were asked to follow a Click Track (metronome) that was set at 90 beats per minute.

4.2.1 Musician Demographic

We asked our musicians about their music playing level, number years of experience, whether they played with others or alone, and their ability to play with a click track. Figures 16 to 19 show the results of the questionnaire.

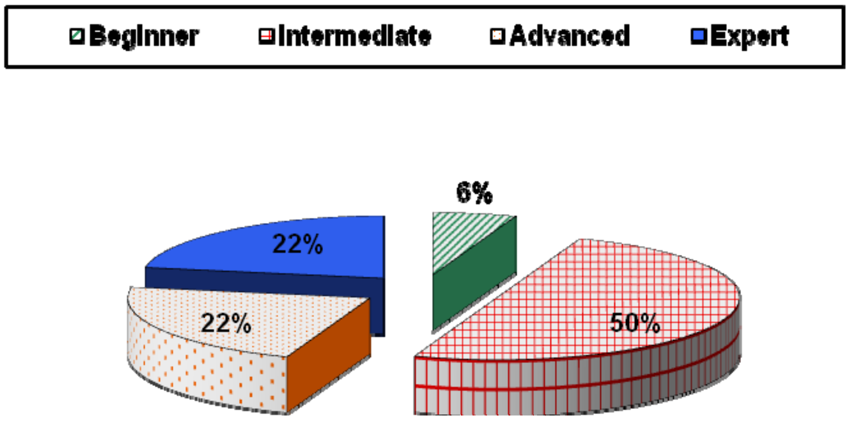


Figure 16 Musical Experience

Musicians answered the level of musical experience by four levels, which are Beginner, Intermediate, Advanced and Expert. 94% of our subjects considered themselves to be intermediate or better.

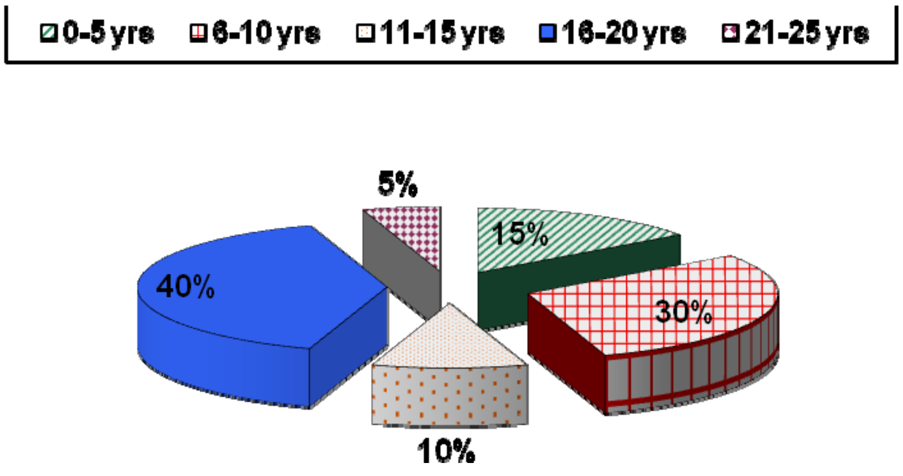


Figure 17 How long have you been a musician?

Musicians answered the level of the number of years of experience by five levels, which are 0-5 years, 6-10 years, 11-15 years, 16-20 years and 21-25 years. 85% of our subjects had more than 5 years of experience.

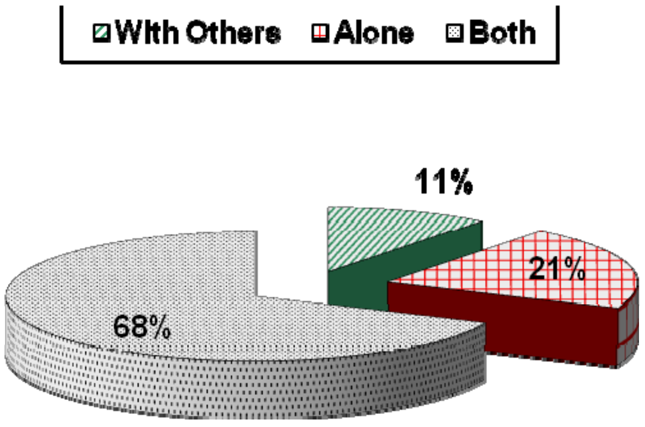


Figure 18 Majority of your playing?

Musicians answered the majority of their playing experience by three levels, which are alone, with others and both. 79% of our subjects had played both alone and with others.

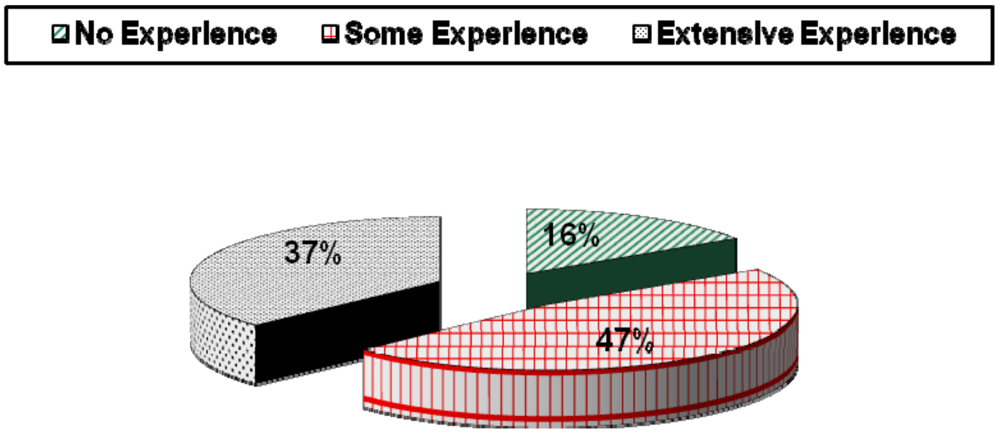


Figure 19 Ability to play with a click track

Musicians answered their ability to play with a metronome (click track) by three levels, which are no experience, some experience and extensive experience. 84% of our subjects had played with a metronome.

Chapter 5

5.0 Analysis and Results

Of the eleven sessions, one was discarded because of an inability to play the piano and an inability to clap to rhythm. Lack of competence was judged subjectively. In the discarded session, either one of the musicians could not play the full piece or the musician kept falling behind because he/she did not remember the keys on the piano.

The audio files were analyzed for tempo consistency at 90 BPM as a function of network delay and ensemble differences between the players were also analyzed as a function of network delay. The clapping audio files were analyzed using Matlab to detect the claps and calculate the time differences between the players. The piano audio files were analyzed using absolute time references in Pro Tools.

5.1 Tempo Consistency

5.1.1 Clapping Tempo

The musician's clapping experimental results were analyzed against his or her subjective results, which was collected with a questionnaire after each trial. The temporal duration of each bar was measured for each of the clapping waveforms.

5.1.1.1 Musicians' Experimental Results

The average tempo consistency as a function of network delay for clapping between two musicians is shown in Figure 20.

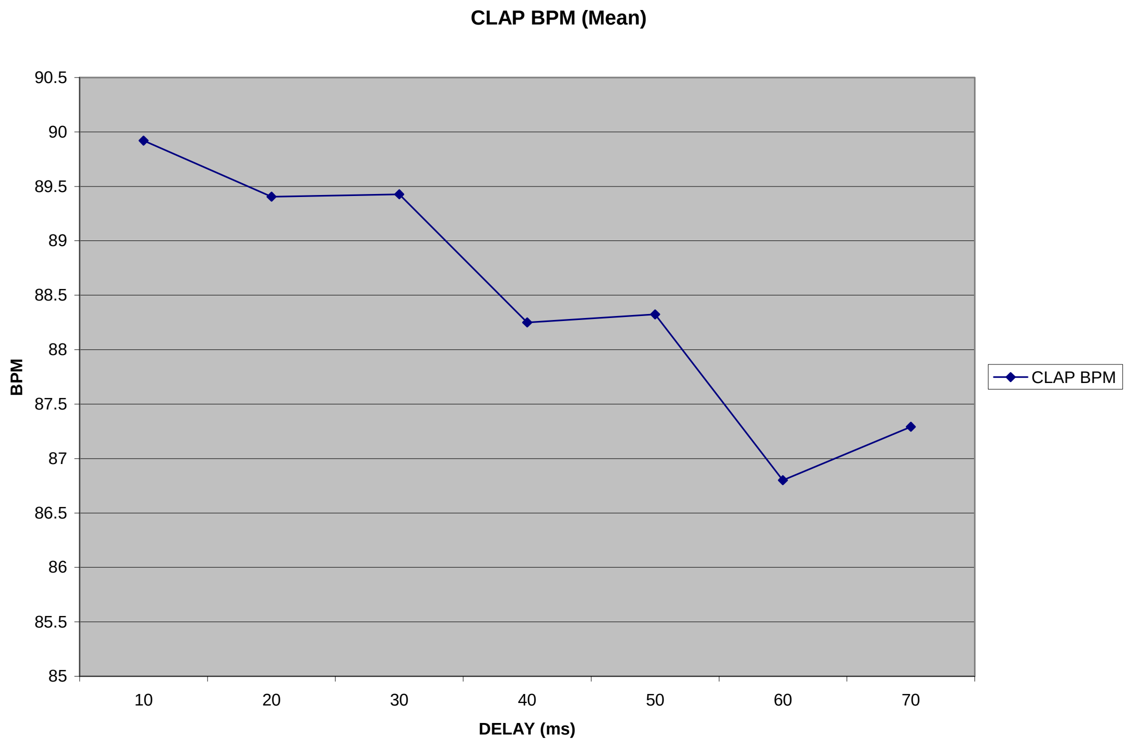


Figure 20 Tempo vs. Network Delay for Duo-Clapping

(*Does not include Application Delay)

It can be seen that the musicians start at 90 BPM and then start to slow down as the network delay is increased. There seems to be a staircase effect, the drop in tempo is large for every 20 ms of network delay.

5.1.1.2 Musicians' Subjective Results

After every trial, the musicians were asked if they felt that they were behind the beat, right on or ahead of the beat. Figure 21 shows the survey results.

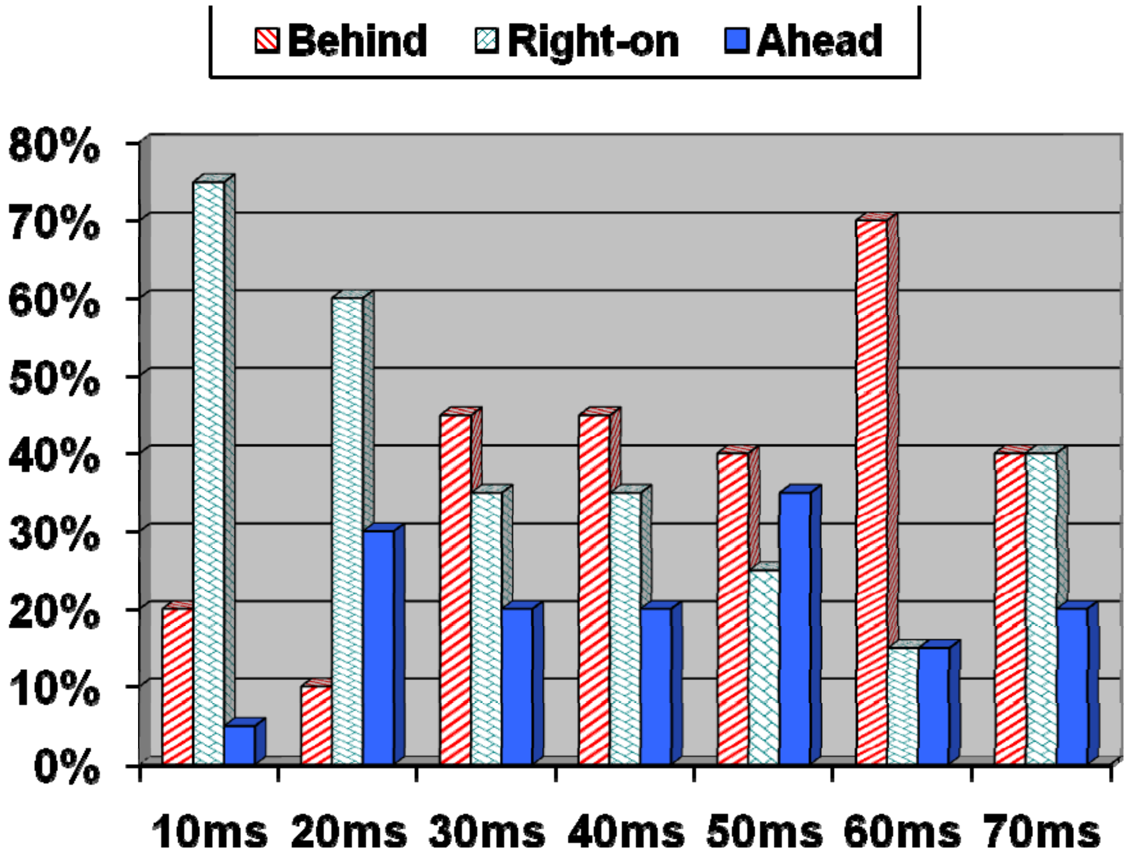


Figure 21 Subjective Results for Clapping
(*Does not include Application Delay)

The results show that the musicians felt that they were right on the beat at least 41% of the time. In most cases the musicians could not distinguish between being ahead or behind but they knew if they were not on the beat. The poor performance could be because it was difficult for them to distinguish between the click track and the other musician's clapping.

5.1.1.3 Experimental versus Subjective

It can be seen from Figures 20 and 21 that as the beats per minute start to drop below 89, the musicians start to feel that they are behind. In this case, the subjective results are consistent with the experimental results.

5.1.2 Piano Performance Tempo

The musicians' piano performance experimental results were analyzed against his or her subjective results, which was collected with a questionnaire after each trial. The temporal duration of each bar was measured for each of the musical waveforms.

5.1.2.1 Musicians' Experimental Results

The average tempo consistency as a function of network delay for the piano performance between two musicians is shown in Figure 22.

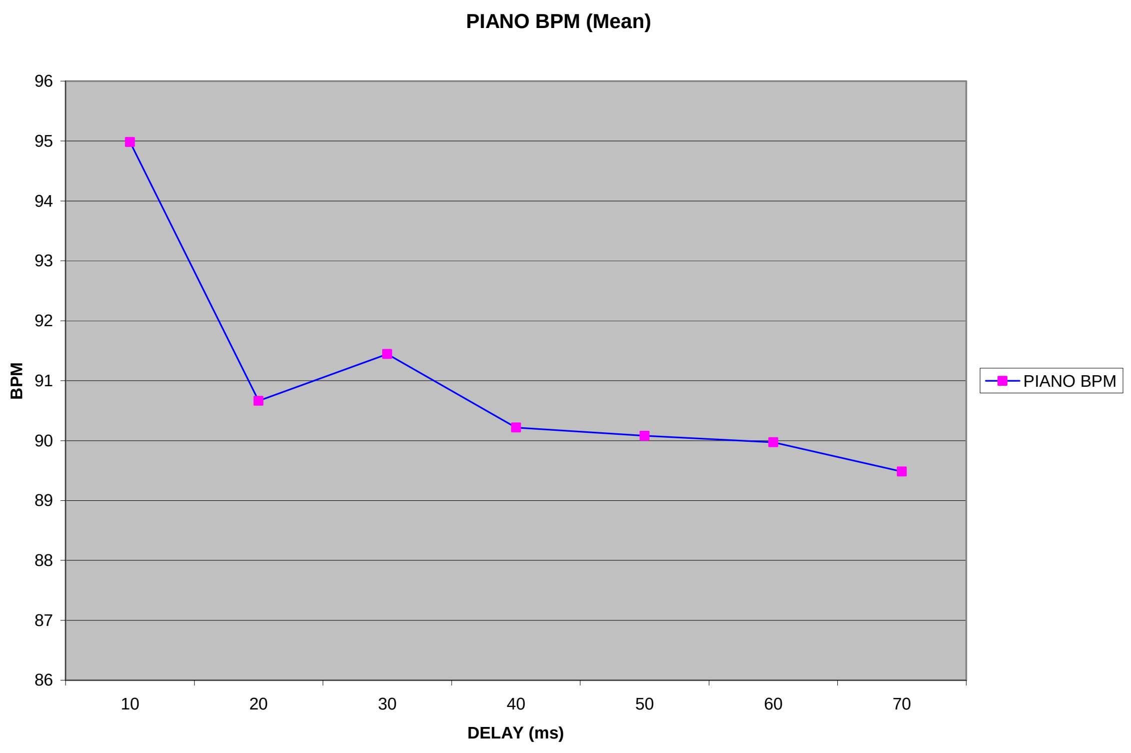


Figure 22 Piano BPM as a function of Network Delay
(*Does not include Application Delay)

It can be seen that the subjects start at 95 BPM and then start to stabilize at 90 BPM as the network delay is increased. The players are playing at a higher tempo at 10 ms because it has been shown that network delays lower than 11.5 ms can accelerate the tempo [1]. But the performance starts to slow down at 70 ms network delay.

5.1.2.2 Musicians' Subjective Results

After every trial, the musicians were asked if they felt that they were behind the beat, right on or ahead of the beat. Figure 23 shows the survey results.

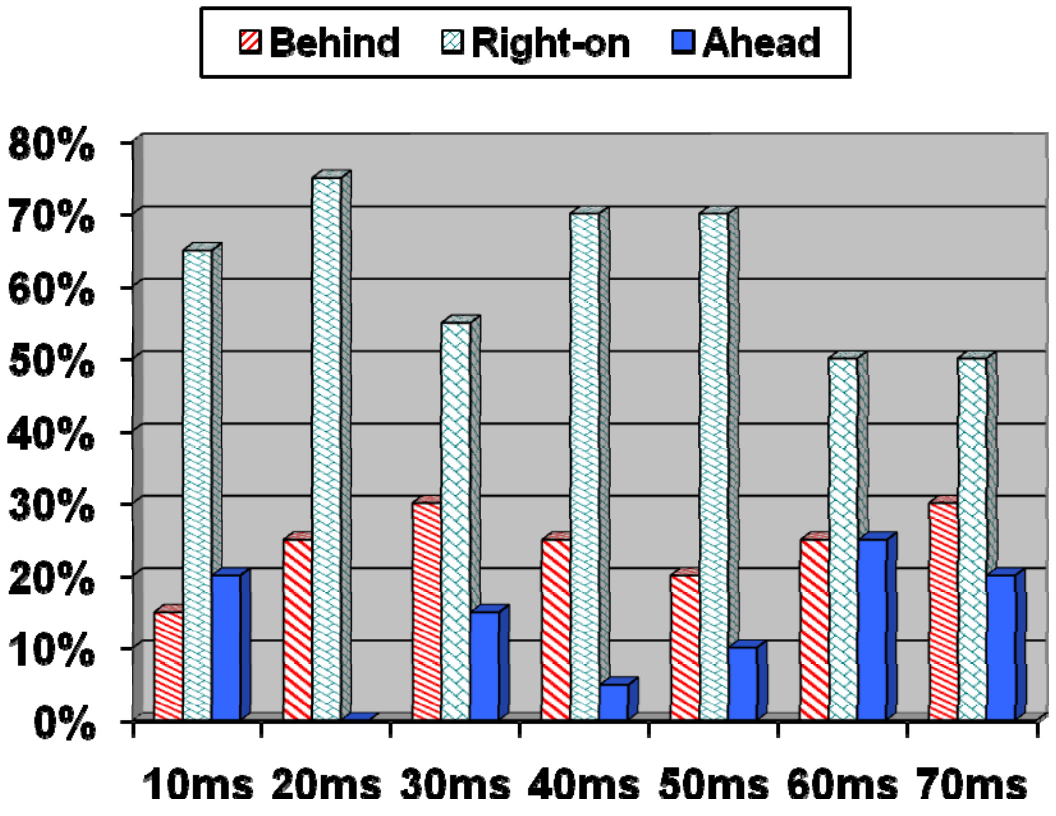


Figure 23 Subjective Results for Piano Performance
(*Does not include Application Delay)

The results show that the musicians felt that they were right on the beat at least 62% of the time. The better performance with the piano can be attributed to musicians being able to listen to the click track while playing the piano and also that playing in unison is easier than clapping in an inter-locking pattern.

5.1.2.3 Experimental versus Subjective

It can be seen from Figures 22 and 23 that the musicians' subjective results are consistent with the experimental results for the network delay range between 20 ms to 70 ms. However, when the network delay was 10 ms, the beats per minute was 95, which is inconsistent with the subjective results. The subjects could have been excited about the fact that there was almost no delay and happen to speed up.

5.1.2.3 Clapping versus Piano Performance

It is apparent from Figures 20-23 that it was easier to play music in unison than clap in an inter-locking pattern over the network with delay. Both the subjective results and the experimental results prove this. The clapping was also hard to perform because the click track kept blending into the claps. Therefore it was tough for the musicians to be synchronized with both the other musician and the click track. A visual click track might be more helpful, so that it would be anticipated by the musicians.

5.2 Ensemble Consistency

5.2.1 Clapping Ensemble Performance

The musicians’ clapping experimental results were analyzed against his or her subjective results, which was collected with a questionnaire after each trial.

5.2.1.1 Musicians’ Experimental Results

The average (mean) of the ensemble differences between the musicians for duo-clapping as a function of network delay is shown in Figure 24.

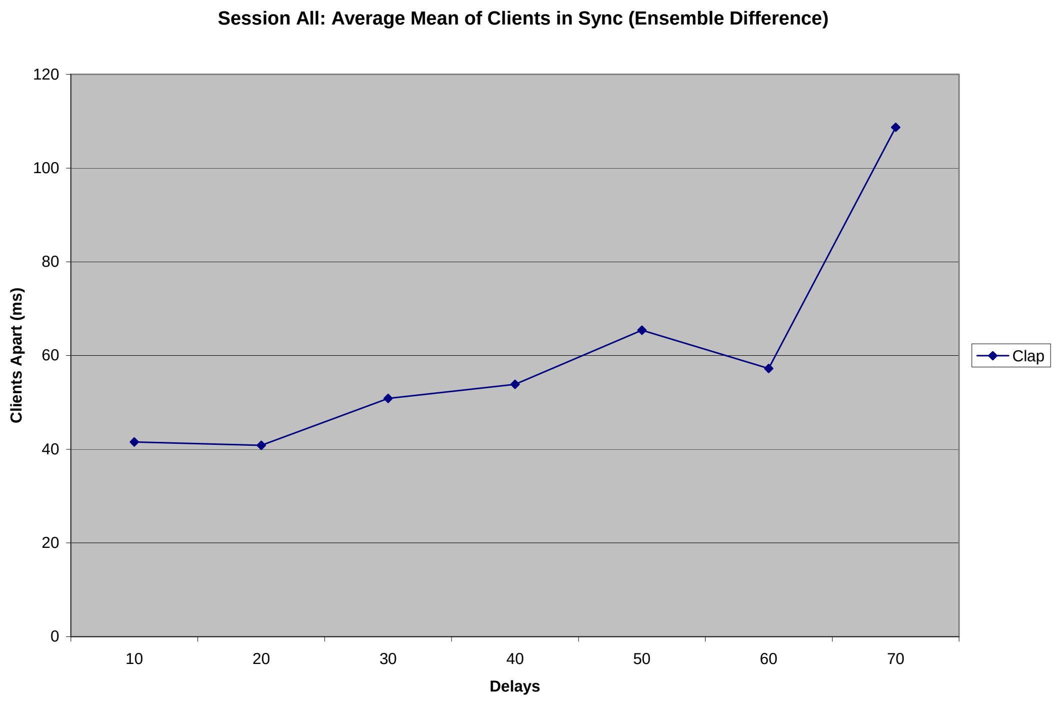


Figure 24 Mean Ensemble Performance for Clapping

(*Does not include Application Delay)

It is clear from the graph that ensemble delay between the players keeps increasing as the network delay is increased. The spread of the ensemble difference values (standard deviation) between the musicians for duo-clapping as a function of network delay is shown in Figure 25.

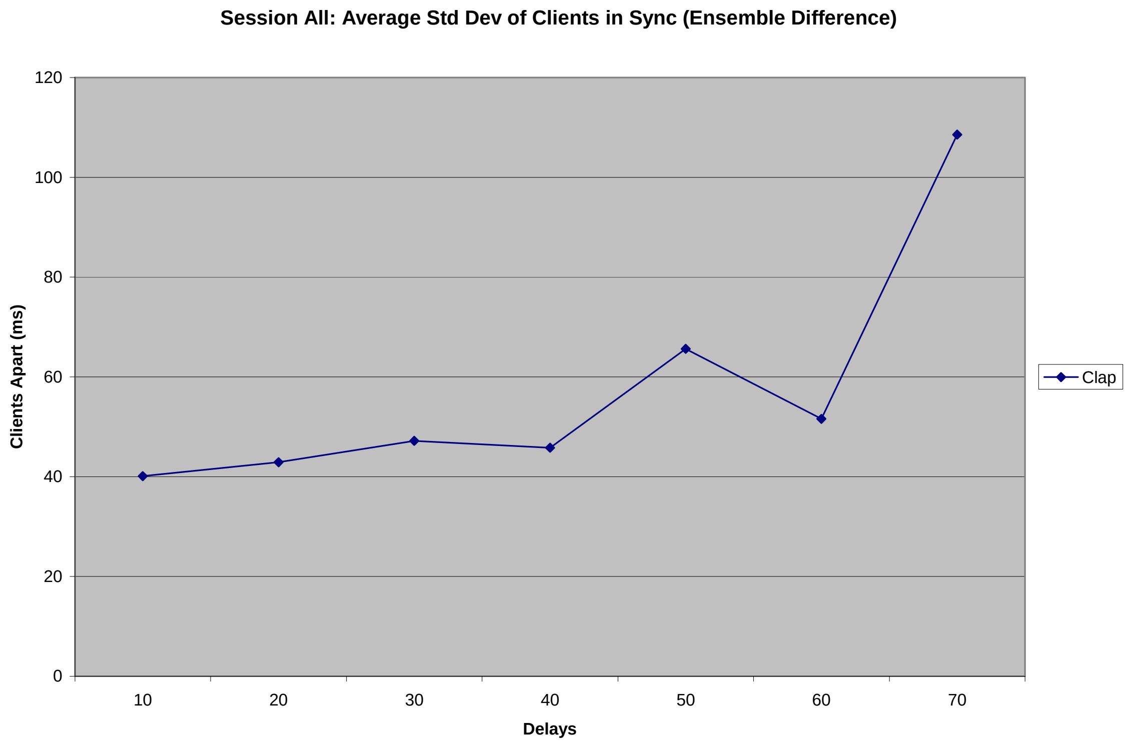


Figure 25 Standard Deviation of Ensemble Clapping

(*Does not include Application Delay)

The standard deviation seems to follow the same trend as the mean ensemble difference.

5.2.1.2 Musicians' Subjective Results

After every trial, the musicians were asked to answer if they felt that they were behind, right on or ahead of the other musician. Figure 26 shows the survey results.

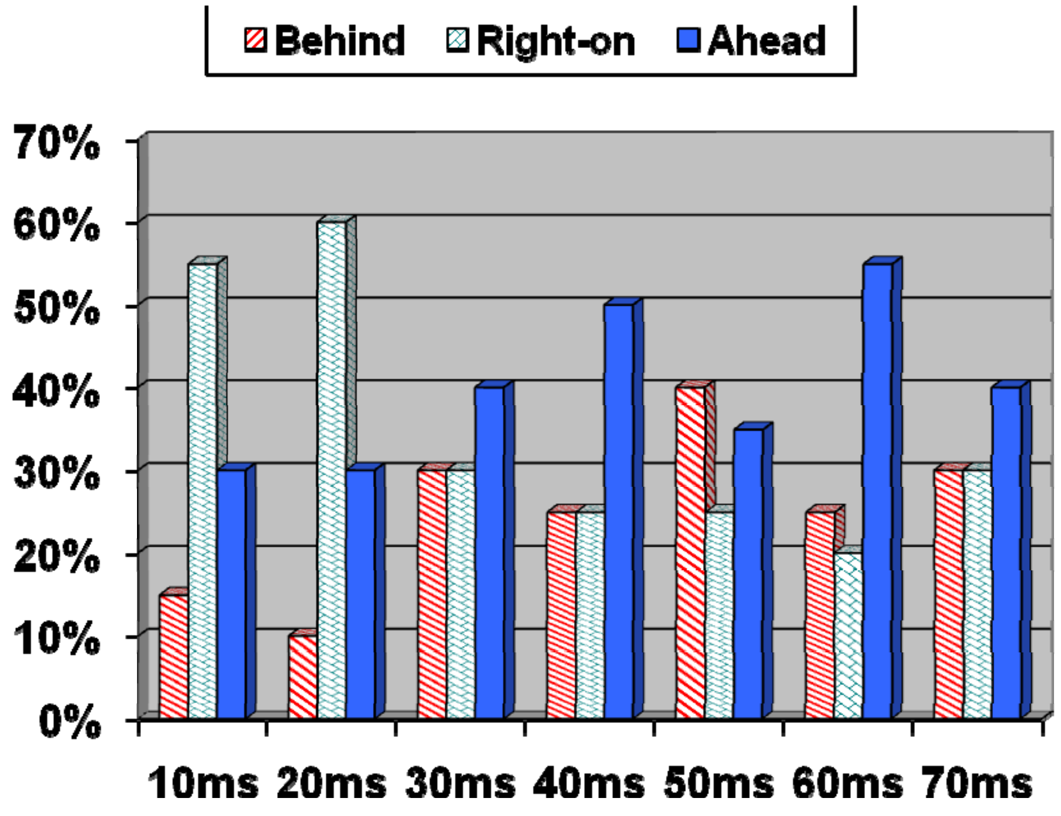


Figure 26 Subjective Results for Duo Clapping Ensemble

(*Does not include Application Delay)

The results show that the musicians felt that they were right-on with the other musician at least 35% of the time. In most cases, the musicians could not distinguish between being ahead or behind but they knew if they were not right-on with the other musician. They feel that they are right-on at 10 ms and 20 ms network delays but with network delays greater than 30 ms they feel that they are either behind or ahead.

5.2.2 Piano Ensemble Performance

The mean ensemble difference between the musicians for “twinkle-twinkle little star” as a function of network delay is shown in Figure 27.

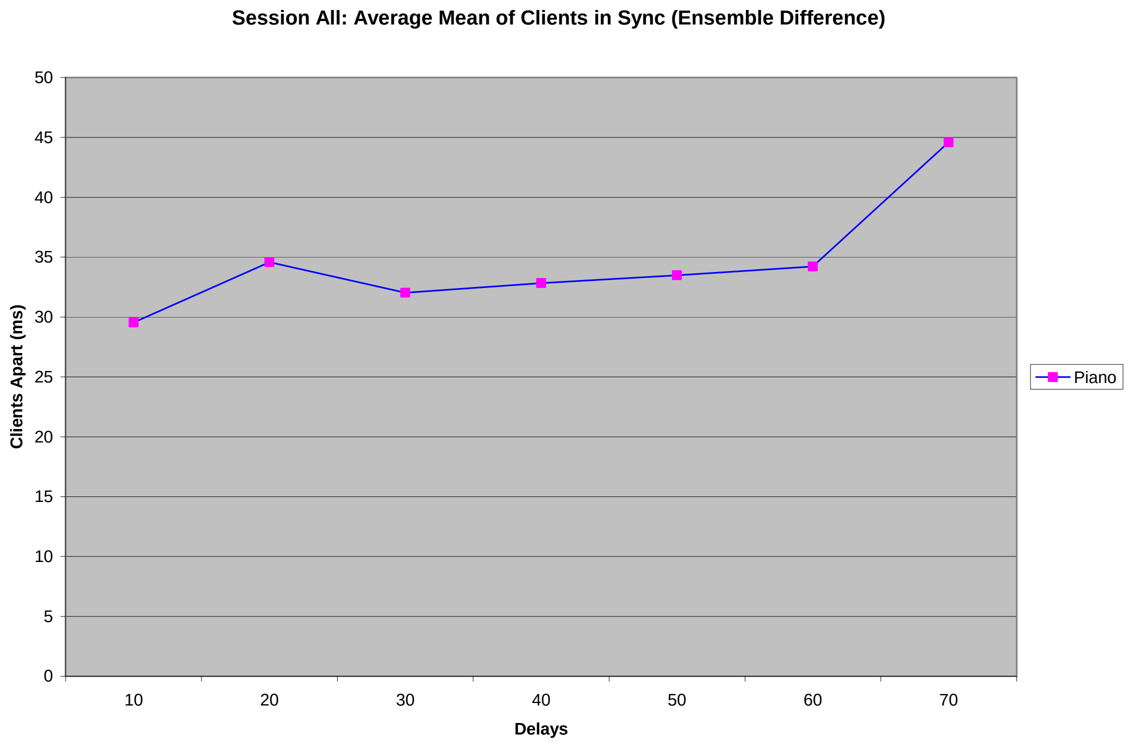


Figure 27 Mean Ensemble Performance for Piano

(*Does not include Application Delay)

The standard deviation of the ensemble difference between the musicians for "twinkle-twinkle little star" as a function of network delay is shown in Figure 28.

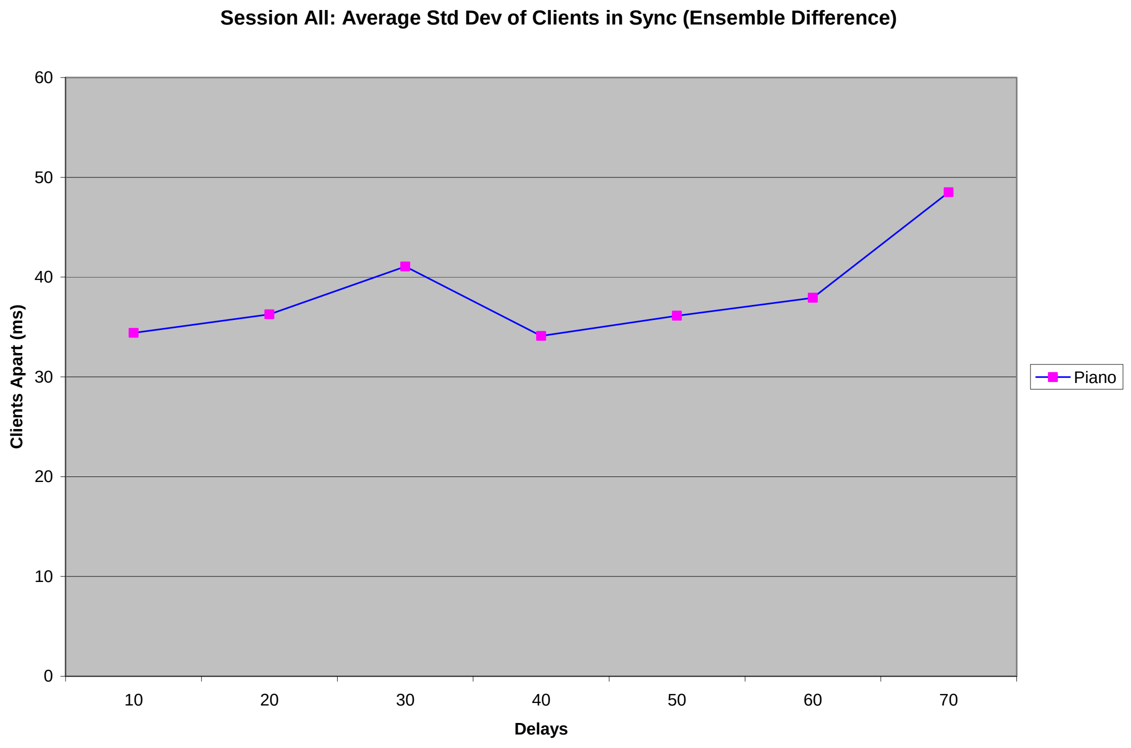


Figure 28 Standard Deviation of Ensemble Piano

(*Does not include Application Delay)

The standard deviation shows a similar trend as the mean ensemble difference between the musicians.

5.2.2.1 Musicians' Subjective Results

The musicians' subjective results for the piano performance are shown in Figure 29.

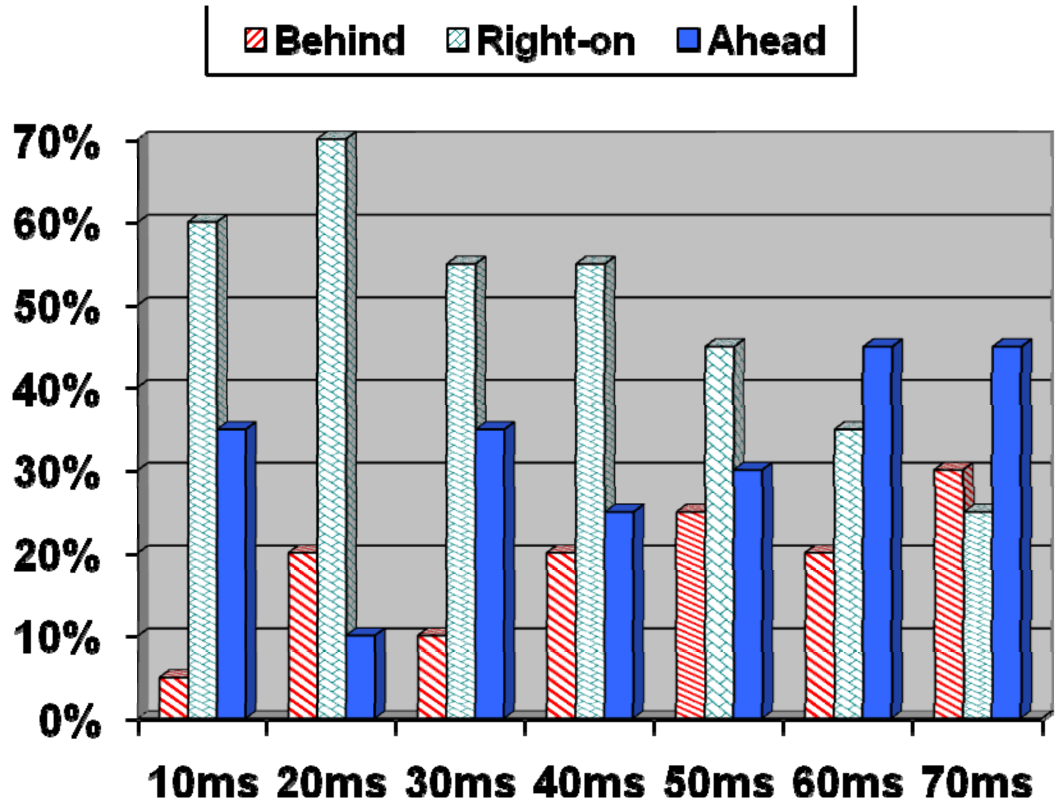


Figure 29 Subjective Results for Piano Ensemble

(*Does not include Application Delay)

The results show that the musicians felt that they were right-on with the other musician at least 49% of the time. In most cases, the musicians could not distinguish between being ahead or behind but they knew if they were not right-on with the other musician.

5.2.3 Comparison between the Duo-Clapping and Piano Performance

It can be seen that the piano performance starts to stabilize at 90 bpm while the duo-clapping has the staircase effect and does not stabilize. The musicians were more comfortable playing the piano in unison at higher delays than clapping with each other.

5.3 Musician Enthusiasm to Jam Online

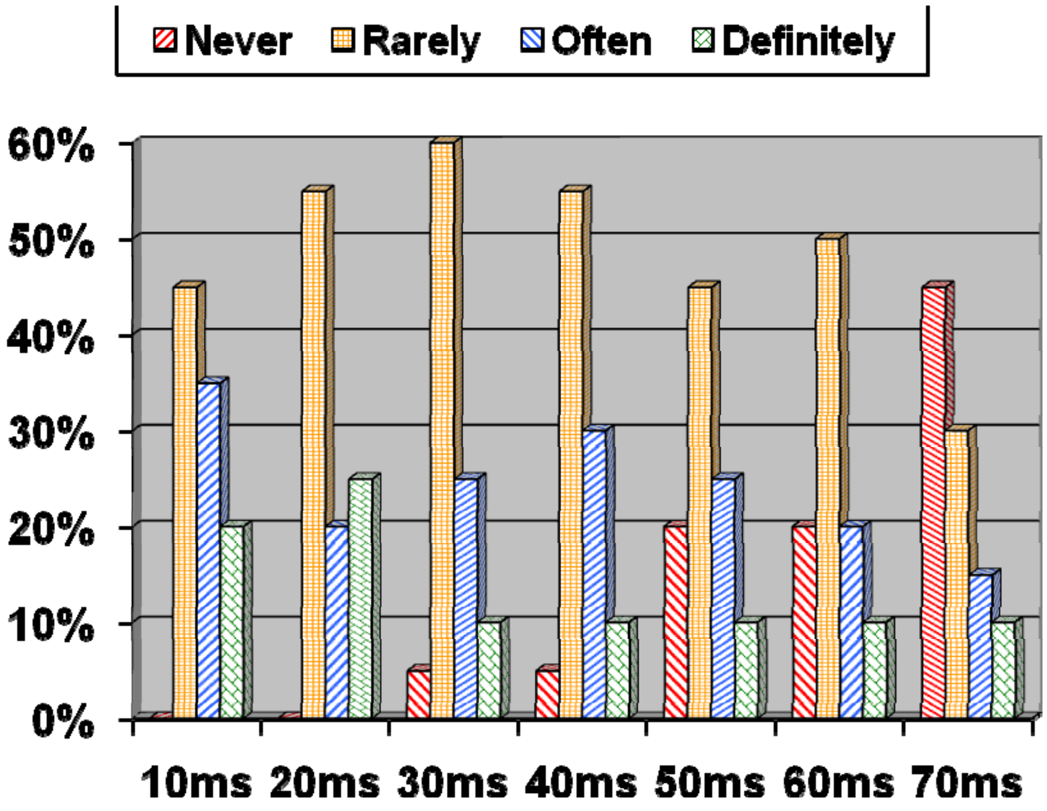


Figure 30 Will you Jam Online? Clapping

(*Does not include Application Delay)

The musicians are likely to play over the internet 64% of the time. At 10 ms network delay, 55% agreed to jamming online but this number dwindled down to approximately 30% as the network delay was increased, as shown in Figure 30. This is consistent with the experimental data collected for the clapping sessions. Figure 20 shows the tempo dropping drastically at network delays greater than 50 ms and Figure 24 shows the ensemble difference to be greater than 50 ms at network delays greater than 40 ms.

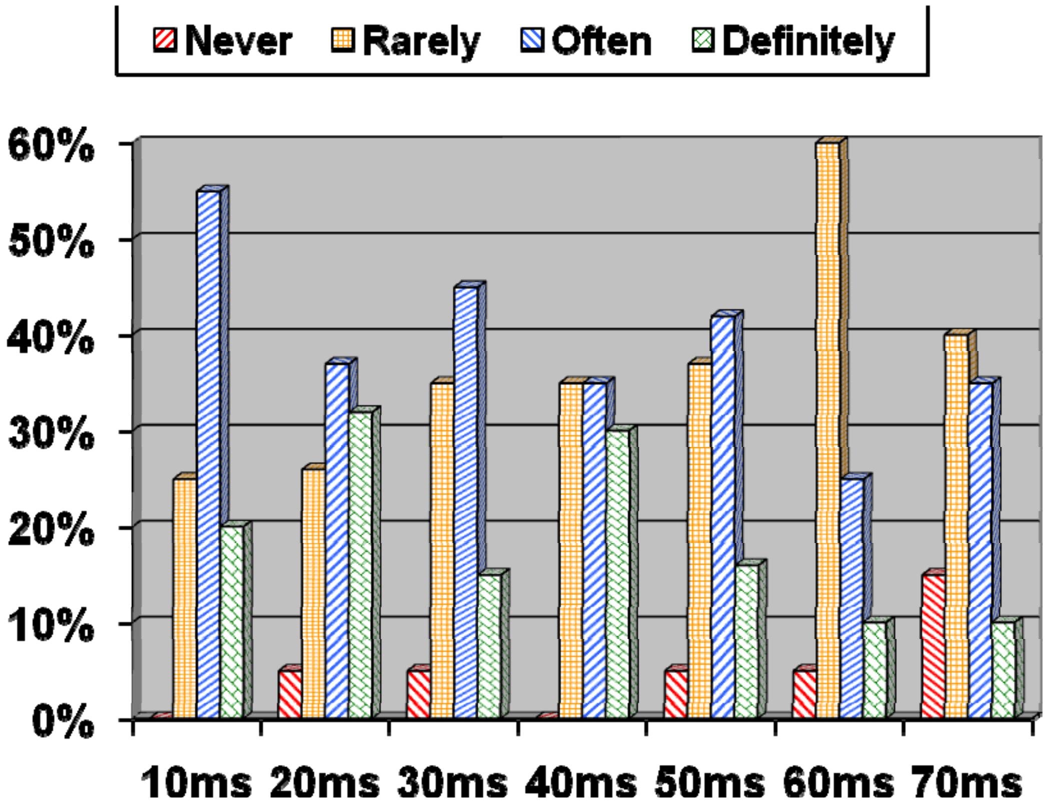


Figure 31 Will you Jam Online? Piano
(*Does not include Application Delay)

The musicians are likely to play over the internet 75% of the time. As shown in Figure 31, at 10 ms network delay, 75% agreed to jamming online and this number dwindled down to only 55% as the network delay was increased to 50 ms. There was a large drop in musicians wanting to jam online at 60 ms and 70 ms. This is consistent with the experimental data collected for the piano sessions. Figure 22 shows the tempo at approximately 90 bpm for network delays from 40 ms to 70 ms and Figure 27 shows the ensemble difference to be less than 35 ms at network delays less than 60 ms.

It can be seen from Figures 30 and 31 that the musicians felt more comfortable playing the piano in unison over the network than perform clapping in an inter-locking pattern. Generally, it is more challenging to clap in an inter-locking pattern than play a musical piece in unison.

Chapter 6

6.0 Conclusions

6.1 Summary

A trace-reading emulator with an accuracy of 1 ms was developed at the price of a regular PC and with a form factor of 12 inches by 8 inches by 8 inches. The emulator was built using Linux, NetEm and TCN. Routing capabilities and a DHCP server were also added using all freeware utilities.

This emulator was used to create a network for online jamming that formed the basis of a subjective test to understand delay tolerance. The results derived from this emulation environment show a very strong pattern of tempo as a function of network delay, and also ensemble time difference between the musicians as a function of network delay.

6.2 Main Contributions

6.2.1 Network Emulator

A single box Network Emulator was developed, with the following features:

- Trace readable
- Routing Capability
- Accuracy of 1 ms
- Emulate multiple network delay profiles for multiple routes simultaneously

6.2.2 Test Methodology

An original methodology was developed to quantitatively and qualitatively measure the musician’s experience with online jamming.

Methodology:

- Two musicians performing together, performing either a clapping session or a piano session with random delay
- Audio trace is analyzed for both tempo and ensemble consistency
- Musicians answered a subjective questionnaire about their experience after each session

6.2.3 Test Results

The live testing of twenty two musicians provided the following quantitative and qualitative results:

- Musicians can tolerate round trip network delays of up to 50 ms when playing a musical piece.
- Musicians can tolerate round trip network delays of up to 30 ms when clapping in rhythm together.
- Round trip network delays greater than 60 ms are detrimental to the jamming experience and musicians are no longer willing to jam online at these delays.
- The tempo was stable during the musical piece over 20 ms to 60 ms range of network delays, but the duo clapping had a stair case effect with large drops in tempo every 20 ms.
- For the musical piece, the ensemble time difference between the players was approximately 30 ms for the network delays ranging from 10 ms to 60 ms, but the duo clapping had a higher ensemble difference of 40 ms and it climbed up to 60 ms as the network delay was increased to 60 ms.
- Network delays of 70 ms were unacceptable for both the musical piece and duo clapping.
- The musicians are more comfortable with a musical piece and are willing to forgive small errors.

The JamNow application has an upper bound application delay of 20 ms. The results did not take into account the 20 ms application delay because the delay variance is unspecified and the delay varies between applications.

6.3 Future Work

Some of the future work derived during the course of this work is outlined below.

- Variable Delay: The delays should be changed from fixed delays to fixed delays with variance. This will model the internet traffic more accurately.

- Trace Files: These sessions should be repeated using ping trace files obtained from existing internet connections. This will model a specific connection from point to point.

- Trio Ensemble: Sessions with three musicians playing together with all three having the same delay and also with each of them having a different delay.

- Capability to Learn: It needs to be seen if the musicians can increase their tolerance to delay with more practice jamming online. This data can be collected over a few weeks by bringing in the musicians and having them play the same music piece.

- Correlation Analysis: Further sessions have to be conducted with new music material that can be analyzed easily for correlation between players. If a player speeds up, does the other player slow down and does this happen at a repetitive rate.

Bibliography

[1] Chafe, C., Gurevich, M., Grace L. and Sean T. Effect of Time Delay on Ensemble Accuracy. 2004. Proceedings of the International Symposium on Musical Acoustics, Nara - Japan (ISMA2004).

[2] Chafe, C., Wilson, S., Leistikow, R., Chisholm, D. and Scavone, G. Simplified Approach to High Quality Music and Sound over IP. 159-163. 2000. Proceedings of the Digital Audio Effects Conference, Verona - Italy (DAFX2000).

[3] Astuti, D. Packet Handling. Seminar on Transport of Multimedia Streams in Wireless Internet.

[4] Yoshida, M., Ob, Y., Yonekura T. A Protocol for Real-Time Remote Musical Session. 2005. IEICE Trans. Inf. & Syst., Vol. E88-D, No.5 May 2005

[5] Barbosa, A., Cardoso, J. and Geiger, G. Network Latency Adaptive Tempo in the Public Sound Objects System. 2005. Proceedings the International Conference on New Interfaces for Musical Expression (NIME 2005); Vancouver, Canada.

[6] Cooperstock, J. and Spackman, S. The Recording Studio that Spanned a Continent.2001. IEEE Computer Society Press. International Conference on Web Delivering of Music (WEDELMUSIC 2001) - Florence, Italy.

[7] Keller, A. Manual tc Packet Filtering and netem. 2006. ETH Zurich.

[8] Keller, A. TCN Trace Control for Netem. Porting the Trace Based Network Emulator RplTrc to the Linux Kernel 2.6 Network Emulator Netem. Semester Thesis. 2006.

[9] Baumann, R., Fiedler, U. RepltTrc: A Tool for Emulating Real Network Dynamics. Swiss Federal Institute of Technology. 2005. TIK Report 231.

[10] Almesberger, W. Linux Traffic Control – Implementation Overview. 1999. EPFL ICA.

[11] Goto, M., Neyama, R., Muraoka, Y. RMCP: Remote Music Control Protocol – Design and Applications. Proc. International Computer Music Conference, pages 446-449, 1997.

[12] Hemminger, Stephen. Network Emulation with Netem. 2005. Open Source Development Lab.

[13] Linux Network Administrators Guide. http://www.faqs.org/docs/linux_network/index.html

[14] Blum, Richard. Network Performance Open Source Toolkit - Using Netperf, tcptrace, NIST Net, and SSFNet. Wiley Publishing, Inc.

[15] Devera, M., Cohen D. HTB Linux queuing discipline manual - user guide. 2002.

[16] Gang et al. TransMIDI: A System for MIDI Sessions Over the Network Using Transis. The Hebrew University of Jerusalem, Israel. 1997.

[17] Chafe, C., Leistikow, R. Levels of Temporal Resolution in Sonification of Network Performance. Proceedings of the 2001 International Conference on Auditory Display, Espoo, Finland, July 29- August 1, 2001

[18] Barbosa, A. Displaced Soundscapes. PhD Thesis, April 2006.

[19] Sarkar, M., Vercoe, B. Recognition and Prediction in a Network Music Performance System for Indian Percussion. NIME07, June 7-9, 2007.

[20] Zimmermann et al. Distributed Immersive Performance. University of Southern California, Thorton School of Music.

[21] Chew et al. A Second Report on the User Experiments in the Distributed Immersive Performance Project. 5th Open Workshop of MUSICNETWORK: Integration of Music in Multimedia Applications. 2005.

[22] Bargar et al. AES White Paper: Networking Audio and Music using Internet2 and Next Generation Internet Capabilities. 1998.

[23] Kurose, J., Ross, Keith. A Top Down Approach Featuring The Internet. Pearson. 2004.

[24] Lazzaro, J., Wawrzynek, J. A Case for Network Musical Performance. 2001.

[25] Greene, K. jam Online in Real Time. Technology Review by MIT. May 25, 2007.

[26] Balliache, Leonardo. Differentiated Service on Linux HOWTO. 2003. http://www.opalsoft.net/qos/DS-20.htm

[27] JamNow User Manual. http://38.113.84.12/client/User_Manual.htm

[28] Lightspeed Audio Labs, http://www.lightspeedaudiolabs.com/