CO Trial
Ex. 403

Before the
COPYRIGHT ROYALTY JUDGES
LIBRARY OF CONGRESS
Washington, D.C.

In the Matter of Mechanical and Digital Phonorecord Delivery Rate Adjustment Proceeding	Docket No. 2006-3 CRB DPRA

EXPERT REPORT OF KETAN MAYER-PATEL ON BEHALF OF NATIONAL MUSIC PUBLISHERS' ASSOCIATION, INC., THE SONGWRITERS GUILD OF AMERICA AND THE NASHVILLE SONGWRITERS ASSOCIATION INTERNATIONAL

1. Introduction and Summary of Conclusions

1. My name is Ketan Mayer-Patel, and I am an Associate Professor in the Department of Computer Science at the University of North Carolina, Chapel Hill. I submit this report to: (1) provide my technical opinion on the topics and questions raised by the Copyright Royalty Judges in the *Order Requiring Additional Information* issued on March 11, 2008 (the "March 11 Order"); and (2) respond to the argument made by the Digital Media Association ("DiMA") concerning the technological processes involved in interactive streaming.

2. Based on my analysis and consideration of the Court's questions, I have reached a number of conclusions about the technological processes by which the digital music services that are participants in this proceeding – namely, RealNetworks,[1]

[1] RealNetworks operates its digital music services under the brand name Rhapsody. In this report, I will primarily use the term Rhapsody to refer to the components of the RealNetworks service and its interface with subscribers.

MediaNet and Napster[2] – offer interactive streaming of sound recordings to their subscribers. First, in the case of each participant's streaming service, a complete copy[3] of the sound recording[4] being streamed is transmitted from a server computer through the Internet using standard transmission protocols that necessarily entail the creation of data packets that together comprise the audio file being streamed. These data packets are transmitted through various computers and routers (sometimes referred to as "nodes") until they reach the destination device (usually the user's computer).

3. Second, the various data packets that comprise the sound recording being streamed are reproduced and stored in a "playback buffer" in the Random Access Memory ("RAM") of the user's computer, so that the entire audio file is eventually re-created in the user's RAM. The user's computer is able to specifically identify this audio file as the file which comprises the sound recording being streamed, and it is capable of being perceived by the user because the computer uses the data in the audio file to play the sound recording for the user.

4. Third, in addition to the RAM copy, a copy of each sound recording streamed by RealNetworks and MediaNet through their respective music services is also created and stored on the hard drive, also known as storing "on disk," of

---

2 I understand that Napster was, but no longer is, a participant in these proceedings. Nevertheless, because I understand Napster to be a member of DiMA, which remains a participant, and because Napster is one of the largest interactive streaming services, I have analyzed Napster's service and technology, and have answered the questions posed in the March 11 Order with respect to Napster as well. I refer to all three collectively as "participant services" for ease of reference.

3 In my report, I use the term "copy" or "copies" to refer to reproductions of sound recordings and musical works, although it is my understanding that the Copyright Act uses the term "phonorecord" (rather than "copy") to refer to such reproductions.

4 Throughout this report, when I use the term "sound recording" or "audio file," I am referring to both the sound recording and the musical work embodied therein.

the listener's computer. Napster also makes such a copy once a particular sound recording is streamed a second time. This disk copy of the audio file is accessible to the listener because it resides on the recipient's computer indefinitely. It is a specifically identifiable audio file that can be rewound, played back and played again by the user after the streaming process ends. I have further concluded that once the interactive streaming process creates a copy of the sound recording on a computer's disk, there is little to no technological difference between that copy and the copy transferred as a limited or conditional download.

2. Background and Area of Expertise

5. I received a Bachelor's Degree in Computer Science and Economics from the University of California, Berkeley in 1992. I received a M.S. and a Ph.D. in Computer Science from the University of California, Berkeley in 1997 and 1999, respectively. I teach courses in Web Programming and Multimedia Networking and focus on research on the following issues: peer-to-peer streaming, 2D and 3D video compression, non-linear media, streaming 3D worlds, and distributed media cacheing. One of my primary topics of interest is the interactive streaming of audio and video, and I have been involved in a number of different research efforts related to streaming in general for my entire career. I am also the Chair of Graduate Admissions for the Department of Computer Science. A copy of my CV is attached as Exhibit A.

6. I am currently a member of the editorial board of ACM Transactions on Multimedia Computing, Communications, and Applications, the flagship technical journal for multimedia computing and communications, with a particular focus on streaming media. I am also co-chair of the Steering Committee for the International Workshop on Network and Operating System Support for Digital Audio and Video

(NOSSDAV), which holds a yearly workshop on cutting-edge multimedia issues. I was awarded the National Science Foundation CAREER Award and the Computer Science Student Association Teaching Award in 2003. I have also served on the programming and organizing committees for over a dozen different multimedia conferences and workshops; I have published approximately thirty articles related to multimedia. Notably, I have published thirteen papers over the course of the fifteen-year history of the ACM SIG Multimedia Conference, considered to be the premier conference in the field of multimedia.

7. I have almost twenty years of experience with multimedia technologies. From May 1991 through June 1992, I worked as a programmer for the United States Department of Agriculture in Albany, California. My work focused on developing custom image processing software for analyzing and manipulating X-ray images of foodstuffs for quality and safety control. From June 1992 through June 1993, I worked as a programmer for the University of California, Berkley. My work focused on developing the first publicly-available software-only MPEG-1 video decoder. The resulting code, known as "mpeg_play," has been downloaded over one million times and provided a platform for conducting research in video compression and video streaming for researchers around the world.

8. In June 1993, I became a Graduate Student Researcher and Graduate Student Instructor at the University of California, Berkeley, where my research focused on multimedia streaming and processing systems. My masters thesis described the design and performance of the Berkeley Continuous Media Toolkit. My dissertation work explored issues involved in processing streaming video to implement video effects

in software using parallel computing techniques. My thesis was entitled "Parallel Software-only Video Effect Processing."

9. I became an Assistant Professor at the University of North Carolina, Chapel Hill in January 2000. During this time, my research focused on multidimensional adaptation, coordinated multi-streaming for 3D tele-immersion, and compression for 3D video. I published articles regarding all of these topics co-authored with the graduate students that I mentored and supported.

10. In August 2005, I was promoted to Associate Professor. Since then, my research has examined peer-to-peer streaming technologies for video and audio on demand. I also continue to work on the problems of coordination for multi-streaming and the development of mechanisms to address the needs of distributed multimedia applications that employ many (*i.e.*, 10's or 100's) different media flows with complex inter-stream semantics and adaptation requirements.

3. Scope and Structure of This Report

11. I have been asked by National Music Publishers' Association, Inc., the Songwriters Guild of America, and the Nashville Songwriters Association International (collectively, the "Copyright Owners") to submit this report to (1) rebut the positions taken by DiMA in this proceeding that "interactive streaming is the playing of a specific sound recording without the creation of an audio file that remains accessible on the client computer beyond the playing of such sound recording," (1/28/08 Tr. at 7:7-7:10 (Laguarda)), and that a sound recording "made available via interactive streaming is perceptible to a listener only contemporaneously with its transmission to that listener" and "cannot be rewound, played back, or played again because the user enjoys it," *Mechanical and Digital Phonorecord Delivery Rate Adjustment Proceeding*, Reply of the

Digital Media Association, Docket No. 2006-3 CRB DPRA, at 5; and (2) respond to the following questions raised by the Copyright Royalty Judges in their March 11 Order: (1) What technological process occurs when a listener selects a particular sound recording to be streamed from the perspective of the service offering the interactive streaming and any third party that facilitates that process? (2) Is an audio file downloaded to the listener's computer? (3) If an audio file is downloaded, is the listener able to access that file in any manner? (4) If the listener is unable to access the downloaded file, what prevents such access? and (5) How does the process for interactive streaming differ from that used for conditional or limited downloads?

12. I have also considered the Copyright Royalty Judges' request for testimony and evidence regarding the distinction between digital phonorecord deliveries in general and those that are reproduced or distributed in a way that is incidental to the transmission of the phonorecord. My analysis discusses the copies of audio files created and stored during the interactive streaming process. From a technical standpoint, there is no distinction between an "incidental" and "general" copy. While I cannot offer a legal opinion on this distinction, I trust that some of my analysis of the streaming process and the copies made during that process may be of assistance to the Copyright Royalty Judges on this point.

13. In this report, I analyze the process by which the participant streaming services offer interactive streaming of sound recordings to a listener, with a particular focus on the copies of sound recordings made during that process and the characteristics of such copies. To that end, I have designed an experiment to test the technologies used by the participant interactive streaming services. I subscribed to each

service, selected a sound recording to be played through their interactive streaming platforms, and studied the results. My methodology for this experiment is set forth in detail in Section 5.1 below.

14. Section 4 of this report provides an overview of interactive streaming in general. In Section 5, I analyze the technological processes of the three participant services, present the results of my experiment, and address the questions raised in the March 11 Order. Finally, Section 6 sets forth my conclusions about the technological process of interactive streaming and the copies of sound recordings resulting from that process.

15. My work on this report has been performed by myself and by graduate students at the University of North Carolina, Chapel Hill working under my supervision. I am being compensated at a rate of $400 per hour for my independent analysis in this matter, and my graduate students are being compensated at a rate of $150 per hour.

16. References to sources cited in this report are contained in the footnotes. Where I have obtained information from a website, I have included a snapshot of the webpage being referenced or the full address of the website. In conducting my analysis and drafting this report I have relied on my extensive knowledge of computer science and experience with multimedia and streaming technology. In addition, a list of the materials I have considered in connection with this report is attached as Exhibit B.

17. The network trace files created through my experiment as well as spreadsheets, plots, and figures derived from my analysis of those traces are contained in

the CD that is attached as Exhibit F. The trace files are compatible with the Wireshark network analysis tool available at http://www.wireshark.org.[5]

4. Overview of Interactive Streaming Technology

18. Numerous music and technology providers offer digital music in a wide variety of forms, the most prominent of which are permanent downloads, limited downloads, and interactive streams. A permanent download, exemplified by a sound recording sold through Apple's iTunes Music Store, is the digital delivery of a sound recording of a musical work without any limits on the number of times or period in which a consumer can play the recording. A limited or conditional download, such as sound recordings frequently sold through music subscription services like Rhapsody and Napster, can only be played by the consumer during a limited period (*i.e.*, during the period in which the consumer has an active subscription) or for a limited number of times. Interactive streaming, the focus of this report, in general terms consists of the digital delivery of sound recordings of musical works, using streaming technology, in response to a consumer's request. For reasons I explain below, from a technological perspective, the interactive streaming music delivery model is not necessarily distinct from the limited download model. RealNetworks and Napster are subscription services that offer each of these forms of music directly to their consumers; MediaNet provides a

---

[5] These files all have names that end in ".pcap." The spreadsheets are compatible with Microsoft Excel and use the standard filename extension ".xls." The data files used to generate the plots are plain text files. These files all end in ".txt." The plot files used to generate the figures are compatible with the gnuplot program available at http://www.gnuplot.info. The plot files all end in ".plot." These plots are included as part of figures generated by the program xfig available at http://www.xfig.org. These files all end in ".fig."

technology and content infrastructure to distribution partners who wish to provide their own branded digital music services offering these forms of music to consumers.

19. This section of my report provides an overview of the technology behind interactive streaming services in general by developing a model of an interactive music service. The purpose of this model is to provide an illustrative context for identifying the components of such a service, discussing how these components interact, and explaining how streaming mechanisms work. This discussion will provide the background information needed to adequately answer the inquiries posed by the Copyright Royalty Judges concerning the specific interactive streaming services participating in this proceeding.

4.1 Client-Server Applications and Network Protocols

20. An interactive music service is a specific example of what is called a "client-server application." In this case, the client is the subscriber to the music service. The client program is the application running on the subscriber's computer (*i.e.*, the Rhapsody music player, the Napster music player, or the MediaNet music player) and is typically provided to the subscriber by the music service at the time of subscription. I will use the term client to refer generally to both the actual human user and/or the music application program running on the user's computer. When a sharper distinction needs to be made, I will use the term client application to refer to the program and the term subscriber or user to refer to the person.

21. The function of the client program is to provide the subscriber an interface to the music service's catalog and to allow the subscriber to indicate which sound recording to play. The client program may allow the subscriber to organize available titles into playlists and/or libraries which can be digitally transmitted to the

subscriber's computer. The client program may also allow the subscriber to explicitly indicate that a particular sound recording should be downloaded and stored locally, resulting in a limited download that is still subject to the subscription agreement. The client program may further allow a subscriber to purchase and permanently download a particular sound recording. In this case, access to a local copy of the sound recording is no longer subject to continued subscription.

22. The other half of a client-server architecture is the server. In this case, the server is the music service. The server's function mirrors that of the client, in that it provides the client with catalog information about available sound recordings, as well as the sound recording itself when one is chosen to be played. For ease of explanation, at this time I will discuss the server as if it were a single program running on a single machine connected to the Internet. In fact, the server's functionality may be spread over a number of different resources located at different places on the Internet, and I will elaborate on this idea later in this section.

23. When the client and server communicate information to each other, they do so by employing a "network protocol" or more simply a "protocol." A protocol is simply a formal set of rules by which two processes communicate. These rules determine the syntax and semantics of the messages exchanged and the expected behavior of either party upon having received a particular message.

24. Protocols are developed in layers (also known as a protocol stack) in which protocols for more complex or application-specific communication services are developed on top of protocols that provide a simpler or more generic communication service. The Internet protocols used by the streaming services I have examined for this

report are layered on top of Transport Control Protocol (TCP), which provides a bi-directional, reliable, in-order connection between two programs running on different computers connected to the Internet and is typically the protocol that is used for streaming processes. TCP establishes a persistent two-way connection between the computers. Each side of a TCP connection can write data to the connection in much the same way as writing data to a file. The data written at one end of the connection is delivered to the other end of the connection reliably and in order. Furthermore, TCP incorporates a well-known TCP congestion control algorithm. This means that TCP will attempt to adjust the speed at which it sends data to the current network conditions encountered.

25. TCP is, in turn, layered on top of a protocol called the Internet Protocol (IP). IP provides the basic operation of hop-by-hop routing of data packets between two computers connected to the Internet. To draw an analogy, one can think of TCP as a shipping company that is responsible for getting freight from point A to point B. The TCP freight service guarantees delivery and can take a sequence of items (in this case datagrams or packets of data that comprise parts of the audio file being streamed) which TCP also guarantees will be delivered in order, and not mixed up with any other items being shipped.

26. While the TCP freight service promises to get its payloads to the destination, the truck drivers who work for these shipping companies do not actually know exactly how to get to point B from point A. Fortunately, at every intersection on the Internet (a "node" or "router"), there is a traffic director (*i.e.*, IP) whom the truck driver can ask for directions. The truck driver tells the traffic director where he is trying

to go, and the traffic director tells him in which direction he should go in order to get closer to his destination. When the truck driver gets to the next intersection, he asks the next traffic director to point him in the correct direction. In this way, the truck driver eventually gets to his destination.

27. This analogy, though somewhat simplified, illustrates the main point that TCP is able to accomplish what it promises by building on IP. In this case, TCP offers its "freight service" (*i.e.*, delivering data packets that comprise audio files being transmitted over the Internet) by relying on IP's "traffic director service" at the various nodes and routers that comprise the Internet backbone. Other protocols that provide even more basic services – communicating a message between two computers directly connected to each other, for example – are layered underneath IP. In general, the protocol stack up to TCP is implemented as part of any modern operating system and providing access to TCP is one of the main functions of the operating system.

4.2 The Technology of Streaming Protocols

28. Having discussed the background on how some key network protocols are defined and implemented, I will now discuss "streaming" protocols and how they operate more specifically. A streaming protocol is characterized primarily by the fact that the client makes use of the information communicated from the server without necessarily waiting for all of the data to arrive before use. Streaming protocols are generally used for multimedia data types like audio and video that a user may desire to have presented quickly. Napster, RealNetworks and MediaNet all implement streaming on top of TCP.

29. Once a TCP connection has been established between the client and server for the purposes of streaming, the server's role is to write (*i.e.*, transmit a copy

of) the audio data to the connection. In doing so, the process necessarily makes intermediate copies of the audio data being streamed to facilitate the transmission process. Since the client is playing the sound recording as it arrives, the server needs to write the audio data to the TCP connection at a minimum rate that is at least as high as the sound recording's encoded rate. In other words, if the sound recording is encoded at a rate of 128 kbits/second, then the server must write the audio data to the TCP connection at a rate that is at least 128 kbits/second or higher. If the server fails to achieve this minimum rate, the user's computer may run out of data during the stream, and the user would notice a break in playing the audio file being streamed.

30. TCP always attempts to deliver data at whatever rate the TCP congestion control algorithm determines is appropriate for current network conditions. In practice, the throughput, which is the rate of data received, achieved by a TCP connection can vary widely during the connection's lifetime due to congestion, and may or may not be consistently higher than the sound recording's encoded rate. I will discuss how a streaming protocol layered on top of TCP might deal with this problem when I discuss the client's role in streaming next.

31. Examining the streaming process from the client's side, the client's role is to read audio data from the TCP connection and play the sound recording. This is accomplished by copying the audio data received from the TCP connection into a "playback buffer." The playback buffer is simply local RAM associated with the client application. The client application then writes the audio data from the playback buffer to the sound driver in order to send the audio data to the speakers. Since the rate achieved by TCP may vary considerably, the client must ensure that it has received enough of the

sound recording before playing begins such that the client does not run out of audio data at any point during playback. The basic idea is to allow the audio data to collect in the playback buffer until the client is reasonably sure that it has enough data to survive the variability of throughput received from TCP. This variability in throughput is sometimes referred to as "jitter" and the playback buffer is also known as a "jitter buffer."

32. During the streaming process, in addition to the copy of the sound recording made in the playback buffer, copies of the audio data also exist within the operating systems of the client and server computers. Once the server writes data to the TCP connection, the underlying implementation of TCP within the operating system at the server must hold onto a copy of the data until it is assured that the client-side TCP implementation has properly received it. Similarly, once audio data arrives at the client, the client-side implementation of TCP must maintain a copy of the data until the higher-level protocol takes possession of the data by reading it from the TCP connection. To summarize, in order to play the sound recording, the client application must necessarily copy the audio file from the TCP connection into RAM. Without a RAM copy, no sound recording could be perceived by the user.

33. In addition to the RAM copy, the client may also write the transmitted audio file to more permanent disk-based storage. This would avoid having to stream the audio data from the music service's servers again if the subscriber chooses to play the same sound recording in the future. The concept of keeping a copy of data speculatively for possible future use is generally known as "cacheing." Cacheing is beneficial to the streaming service because it lowers the bandwidth requirements placed on the streaming service's servers, and thus lowers their cost of doing business. In the

context of a subscription-based music service, when the client plays the sound recording again, the client application need only reauthenticate the subscriber's permission to play the sound recording (*i.e.*, that the subscriber has an up-to-date subscription) and the sound recording can then be played from the cached copy rather than re-streamed from the server.

34. Subscription music services, including those offering interactive streaming and download capabilities, anticipate that subscribers will listen to the same set of sound recordings repeatedly and provide them with the tools to do so, in particular through the ability to create libraries and playlists. In addition to allowing users to pick sound recordings to stream, the services also allow users to choose sound recordings specifically to be downloaded to local storage, resulting in limited downloads. These explicitly downloaded copies of the sound recording are still subject to terms of subscription and access to them still requires reauthentication, but they may be more clearly labeled or easily located within the application's music folders. Functionally, however, there is little difference between playing an explicitly downloaded limited download stored on the computer and playing the implicitly downloaded cached copy of a sound recording that was transferred to the computer through the streaming process. In each case, the client receives the request for a particular sound recording and locates the specific audio file to play from the computer's hard drive.

4.3 Digital Rights Management and Encryption

35. It is important to note that music streaming services typically use some form of digital rights management (DRM) in order to protect the audio data from unauthorized access. A major component of any DRM system is key-based encryption. With key-based encryption, data is encrypted using a "key." The key is simply a specific

string of bytes. Generally, the longer the key is in length, the greater the security provided. The key must be known to any party that wishes to decrypt the data. The algorithms used to calculate the encrypted data must ensure that the key cannot be easily guessed from examining the encrypted data. While encryption and encryption technologies can be a complex subject, for the purposes of this discussion, this simple description of key-based encryption will suffice.

36. In order to decrypt the audio data, the client requires the appropriate decryption key. The server provides this decryption key to the client by encrypting the decryption key with yet another key that only a legitimate and authorized version of the client application should have access to. The client application may decrypt the audio data as it is received from the TCP connection. In this case, the audio data in the playback buffer, the local RAM associated with the client application, will be unencrypted. Alternatively, the client application may decrypt the audio data as it is being played. In this case, only portions of the audio data are decrypted as each portion is sent to the sound driver (*i.e.*, the interface to the speaker provided by the operating system).

4.4 Content Distribution Networks

37. In the discussion above, I have described a client-server model that has treated the server-side as if it were simply implemented as a single program running on a specific computer. More realistically, the server's functionality is spread out over a number of different programs running on many different machines. For example, one particular server program may handle client requests for meta-information about the music available (*i.e.*, album listings, cover art, artist bio's, etc.) while a different server program may handle the actual streaming of a specific sound recording. Furthermore,

this functionality may be replicated on a number of different computers in order to be able to handle a large number of simultaneous clients. Replicating functionality necessarily entails creation of one or more copies of the content at issue.

38. In particular, an interactive music service may choose to employ a content distribution network ("CDN") in order to improve streaming performance for its customers. A CDN is a type of Internet business which helps other businesses distribute data to clients that may be geographically dispersed around the country or world. Well-known examples of CDNs include Akamai and Limelight.[6] The basic idea underlying a CDN is to establish server resources (*i.e.*, computers and associated storage) within the infrastructure of many different Internet Service Providers ("ISPs") all over the world. Businesses that wish to distribute data to a large number of clients can contract with a CDN to replicate this data onto these server resources. To ensure that the audio data is delivered quickly and efficiently, and thus facilitate its transmission, copies of the audio data are necessarily created and distributed to the CDN servers used by the interactive music service. Therefore, when a client requires access to a particular data file, the request is redirected to the "nearest" CDN server resource that contains a replicated copy of the data. In this context, nearest refers to distance within the Internet and not necessarily geographic distance, and the music service may choose to rely on the third-party CDN to provide the benefits of "nearby" access for audio data and facilitate the streaming process. In this case, the audio data offered as sound recordings by the music services must be replicated on each of the CDN servers.

---

[6] These services can be reached online at http://www.akamai.com and http://www.limelight.com, respectively.

5. Analysis of the Interactive Streaming Technologies Used By RealNetworks, Napster and MediaNet

5.5 Experimental Framework

39. In order to analyze how RealNetworks, Napster and MediaNet stream sound recordings to their subscribers, and whether each service creates copies of those sound recordings as a result of such streaming, I designed an experiment, detailed below, to examine the technological processes behind each music service. We used a network trace collection program called Wireshark to capture all of the interactions between the client application downloaded on the user's computer (*i.e.*, the Rhapsody player, the Napster player, or the MediaNet player) and each music service, respectively. Each of these client applications is a software application that is specific to that music service and is provided to the user when the user selects and pays for a subscription to the music service.

40. Wireshark is a program that can be used to examine and copy all information being sent to and from a computer.[7] In addition to making a copy of the network packets sent and received, Wireshark saves timing information about when these packets are sent and received. Wireshark can save all of this information in a file known as a "network trace" for later analysis. Wireshark can also read previously saved network trace files, and it provides a number of different tools for analyzing those files.

---

[7] Wireshark is a free program that can be downloaded from http://www.wireshark.org. The program has been under continuous development since 1998. We used Wireshark version 0.99.8, which was the most current version of Wireshark available at the time these experiments were performed. The lead developer of wireshark is Gerald Combs, and the project benefits from dozens of contributing authors. The program has been recognized as the best network protocol analysis tool available. *See* "Best of Open Source Networksing," Infoworld, dated September 10, 2007, located at http://www.infoworld.com/article/07/09/10/37FE-boss-networking_1.html.

41. For each music service, we took the following steps to produce a network trace file:

1. Using a computer under my direction and the Internet, we logged on to the music service's homepage (*i.e.*, www.rhapsody.com, www.performerdigital.com, and www.napster.com).

2. We selected one of the music subscription service packages provided by each service that offered an interactive streaming option and provided payment information in the form of a credit card.

3. We downloaded and installed the appropriate player application provided by the music service onto the computer (*i.e.*, the Rhapsody player, the MediaNet player, and the Napster player).

4. We then set up Wireshark on the computer to record all network activity, noting the time at which Wireshark was enabled.

5. We started the player application for the particular service being analyzed, noting the time.

6. We then selected a particular sound recording to be streamed on the computer, again noting the time.

7. We allowed the sound recording to play to completion, noting the duration of the sound recording.

8. Then we streamed the same recording a second time, using the same process just described.

9. To test what happened when the initial Internet connection was closed, we exited the player application for the music service, restarted the player application in order to reconnect to the service, and then streamed the same sound recording for a third time as before.

10. We exited the player application for the music service again, and closed the connection to the Internet.

11. We stopped Wireshark from recording further network activity.

12. We saved the network activity just recorded to a trace file for later analysis.

42. Subsequent analysis of each network trace file allowed us to determine whether the streaming of each sound recording resulted in the creation of a

secondary cached copy of the sound recording and, if so, where the copy resides and whether a copy of the sound recording was saved to disk-based storage. This analysis was also done using Wireshark, as well as other tools available for reading and interpreting the network trace files described in more detail below, according to the following procedure:

1. We identified all TCP network connections in the trace file that were created by the player application on the computer. (As discussed above, TCP is a generic, reliable, first-in-first-out connection between the computer and the music service's server computers.)

2. We extracted for each of these connections information about the music service's server computer through which the connection was made, including the name of the business entity that administers that server computer.

3. We extracted for each of these connections the timing information concerning when the connection was made and its duration.

4. We extracted for each of these connections statistics about how much data was transferred between the music service's player application on the computer we were using and the music service's server computer.

5. We tabulated the information collected in steps 2-4 into the tables attached to this report as Exhibits C, D, and E. Each row of the table contains information about one particular connection observed in the network trace, in order based on the time it was first observed in the network trace. The column titled "IP Address" provides the Internet Protocol (IP) address of the server machine.[8] The column titled "Business Entity" lists the business entity that has administrative control over the IP address involved in each connection.[9] We used the identity information in these two columns to ascertain whether or not the music service was using a CDN. The time that the first packet of the connection was observed relative to the beginning of the experiment is listed under the heading "Relative

---

[8] Every computer connected to the Internet is assigned an IP address, which serves as the computer's identity on the Internet. This address is comprised of a sequence of 4 numbers between 0 and 255 and is typically written in "dot notation" with each number listed in order separated by a period or dot.

[9] This information was obtained by using a web-based tool located at http://network-tools.com. The web page at that location provides a way to look up the associated business entity given a particular IP address.

Start." The duration of the connection in time is listed under the heading "Duration." Statistics about the data transferred during each connection is provided in the next two columns. The total number of bytes transferred from the music service's server to our computer is listed under the heading "Bytes Received," and the average throughput in kilobits/second is listed under the column heading "Average Data Rate."

6. We then cross referenced the time recorded when the sound recording was played during the trace file capture procedure with the time recorded for each network connection in order to determine which connections were made during each of the three instances when the sound recording was played and to identify exactly which connection(s) contained audio data.

7. For illustrative purposes, we also constructed a graph plotting the rate at which data is transferred from the servers as time progressed during the capture procedure. The x-axis represents time from the start of the experiment in 1-second intervals and the y-axis represents the amount of data transferred in that interval. On this same graph, we demarcated the times noted during the capture procedure when the sound recording was played by three vertical lines.

43. From this analysis we expected to be able to determine whether or not an additional cached copy of the sound recording we streamed was made, where it was made, and whether or not this copy was stored on the computer's hard drive. Sound recordings for streaming are typically encoded at bitrates ranging from 100 – 200 kilobits/second (kbps).[10] A kilobit is a unit of memory space (similar to a megabyte) and a second is a unit of time. A kilobit is typically defined as 1024 bits (or equivalently, 128 bytes). Therefore, the kbps value signifies how many kilobits of space each second of a sound recording takes up. For example, a four minute sound recording may be encoded at 192 kbps. Since we know the length of the sound recording that was played, by multiplying the length of the sound recording (four minutes or 240 seconds) by the

[10] Review of Napster 4.0, PC Magazine, dated October 17, 2007, located at http://www.pcmag.com/article2/0,2817,2200712,00.asp; Review of Rhapsody (Fall 2007), PC Magazine, dated October 30, 2007, located at http://www.pcmag.com/article2/0,2817,2209710,00.asp.

encoding rate (192 kilobits per second) we can estimate approximately 46,080 kilobits of data should be transferred as audio data.

44. This experiment was designed to reveal a number of details about the streaming technology used by each music service. First, through the correlation of the time information in step 6, we would be able to identify the precise connection through which audio data was actually delivered by looking for a connection that transferred the appropriate amount of data, given the sound recording length and bitrate, just after the moment when we played the sound recording for the first time.

45. If we did not observe a connection transferring a similarly appropriate amount of data when the sound recording was streamed a second time, we could deduce that a cached copy of the sound recording must have been made on the computer hard drive in addition to the RAM copy made as a result of the first stream of the sound recording. The data transfer would not occur again because the cached copy could be used to play the sound recording the second time.

46. If we then did not see a connection transferring an appropriate amount of data when the sound recording was played for the third time, despite having exited and restarted the player application on the computer we used, we would be able to deduce that the copy that was made was not just stored in RAM for prompt playback but must also have been stored on the hard drive of our computer and played from there. If we found evidence that a copy of the sound recording was made on to disk, we planned to attempt to locate exactly which file in the computer contained the copy.

47. In the following subsections, I will present the results of running this experiment on each of the services and address the Court's specific questions in regard to each service.

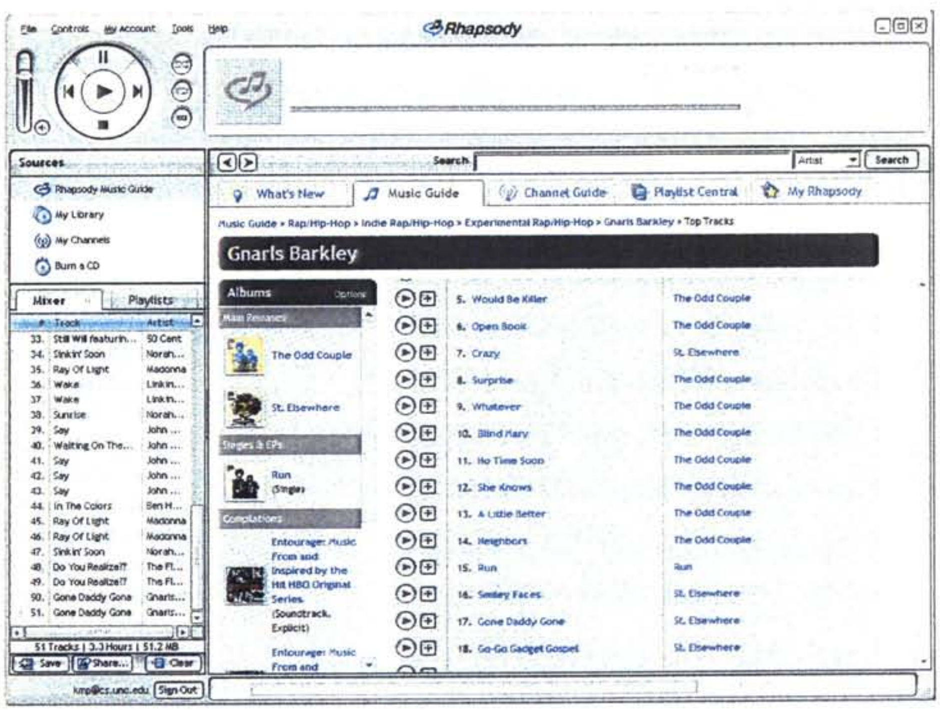
5.6 RealNetworks' Streaming Technology

48. Rhapsody, the subscription service offered by RealNetworks, offers three different subscription plans - "Free Account," "Rhapsody Unlimited," and "Rhapsody To Go." The Free Account subscription allows a person to play 25 sound recordings of their choosing per month without having to subscribe. Rhapsody Unlimited provides unlimited interactive streaming to all 4 million-plus sound recordings in the Rhapsody library. Rhapsody To Go includes the ability to transfer sound recordings to compatible portable music player devices. In order to conduct this analysis, we subscribed to the Rhapsody Unlimited service.

49. Figures 1 and 2 show screenshots of the Rhapsody player application. The Rhapsody player application provides the listener with a number of different tools for selecting sound recordings, including guides organized by genre and artists, playlists organized by other subscribers, and "channels" which contain sound recordings organized by theme or style. Some of these features can be seen in Figure 1. In Figure 2, a listing of sound recordings from a particular artist is shown. To play a particular sound recording, the user presses the play button next to each title.

Figure 1

Figure 2



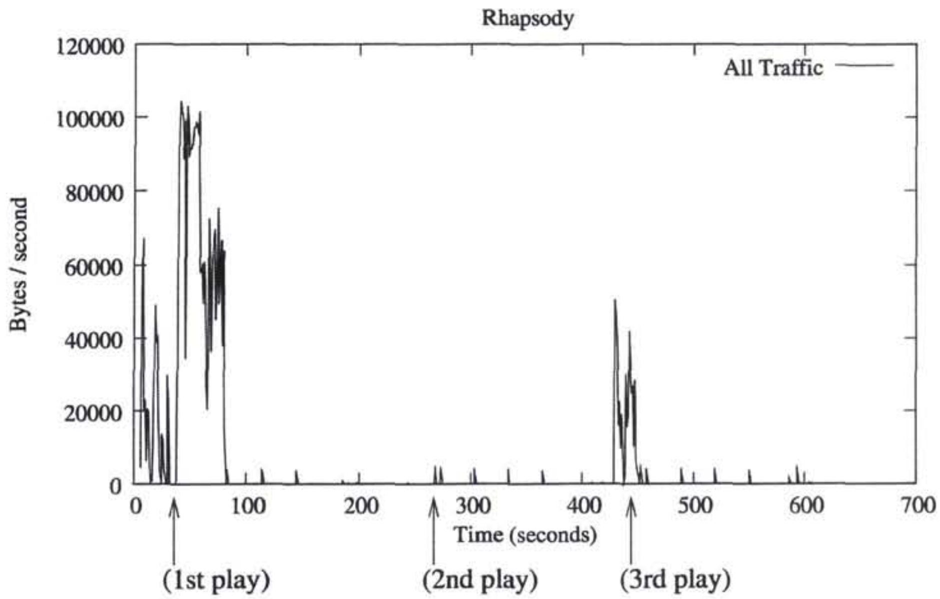
50. The results of the experiment described above as applied to the Rhapsody service are charted in the table attached as Exhibit C. During our experiment, we used "Gone Daddy Gone" by the artist Gnarls Barkley.[11] The playing time for this sound recording is 2 minutes and 28 seconds. During the time in which we streamed "Gone Daddy Gone" three times, a total of 21 TCP connections made between the Rhapsody player application on our computer and the Rhapsody service's server computers are listed.[12] These connections communicated to 5 distinct server computers administered by three different business entities: RealNetworks, Limelight, and Akamai. Limelight and Akamai are two well-known CDN providers with whom RealNetworks has likely contracted to provide CDN services as part of implementing the Rhapsody service. Figure 3, the illustrative graph constructed in step 7 of our experimental procedure, shows the throughput (again, the rate of data received) by the player application over the course of the experiment.

---

[11] While the data represented in the tables below track our analysis of the streaming of "Gone Daddy Gone," we confirmed our results by streaming and analyzing a number of other sound recordings as well. Similarly, while the results reported for MediaNet and Napster focus on the streaming of "Take Me Out" and "Float On," respectively, we checked our results by streaming a range of other sound recordings from each of those services.

[12] Almost all of these connections have nothing to do with transferring audio data but instead represent other types of interactions between the Rhapsody player application and the Rhapsody service. For example: logging in the subscriber, authenticating the account, retrieving song listings, retrieving images such as album cover art for display within the player interface, etc.

Figure 3



51. As described above, we were then able to analyze this data to determine when audio data was actually transferred from the server to the computer we were using, and to deduce whether or not a copy of the sound recording was made on our computer. Our experiment revealed the following with respect to the Rhapsody music service:

(a) When the sound recording was played for the first time, the audio data was delivered by Connection Number 9. The sound recording selected chosen to be played is 148 seconds in length. Connection 9, which began immediately after the sound recording was chosen to be played, was 2,472,067 bytes. This represents a sound recording encoding rate of 133 kilobits/second for the length of the sound recording. This is a reasonable bitrate for audio data. Furthermore, no other connection delivered an appropriate amount of data, given the sound recording's length, at an appropriate time in the experiment.

(b) At a minimum, a copy of the sound recording was stored in the RAM of our computer. As described above, when we selected the sound recording to be streamed, Rhapsody servers created a copy of the audio file containing the sound recording selected. This copy of the audio file was sent through the Internet in data packets using the TCP protocol, and was buffered, or stored, in the RAM of the computer. Once the Rhapsody player application determined that a sufficient amount of audio data had been copied, the audio data in RAM were sent to the computer's speakers. Thus, the evidence indicates that streaming an audio file containing a sound recording entails creation of a full and complete copy of the audio file that comprises the sound recording in the computer's RAM. This copy is capable of being perceived, and is, in fact, perceived as the audio file in RAM is sent, by the player application, to the computer's speakers.

(c) In addition, a cached copy (again, meaning a copy of data that is speculatively kept for possible future use) of the sound recording must have also been made, either in RAM or on the computer's disk, since there was no connection evidencing the transfer of an appropriate amount of data during the second attempt to stream the sound recording. As I described earlier, we knew that if no connection during the second instance of streaming the sound recording transferred an appropriate amount of data, we could deduce that a cached copy of the sound recording must have been made during the first instance, and used to play the sound recording again. This can be seen graphically in Figure 3. The portion of the graph starting at the mark labeled "2nd play" indicates the throughput levels when the sound recording was streamed the second time. These levels are much lower than those of the first stream, indicating that data was not being transferred from the server. If audio data had been transferred again, this portion of the graph would look similar to the portion of the graph marked as "1st play."

(d) Additionally, we confirmed that a cached copy of the sound recording made during this process was, in fact, stored on the computer's hard drive (in addition to the copy made in RAM) rather than simply in RAM, because when we played the sound recording for the third time, having exited and restarted the Rhapsody player application between the second and third times we streamed the sound recording, we again did not detect a connection delivering the appropriate amount of data. This too can be seen graphically in Figure 3 as relatively low throughput levels starting at the mark "3rd play."[13] Because a copy stored only in RAM would not have been available after exiting and restarting the application, and because no additional transfer of audio data occurred during the third playback, a copy must have been made and stored on disk during the first stream, which was then used to play the sound recording the third time.

52. Upon further investigation, we were also able to determine where on disk Rhapsody's service stores cached copies of the sound recording. These copies appear to be stored in a file named "radfile.rcf" and are located within the subdirectory where the Rhapsody player application is installed. We discovered this by conducting the following secondary experiment:

1. We started the Wireshark tool to create a network trace.
2. We used the Rhapsody player application to play a sound recording that we had never played before.
3. We exited and restarted the Rhapsody player application.

---

[13] The throughput activity that occurs just before this mark is associated with the Rhapsody player application being restarted, initialized, and authenticating the subscription.

4. We streamed the same sound recording again.

5. We exited the Rhapsody application once more; we stopped Wireshark and saved the network trace.

53. Using the same analysis described above, we examined the network trace captured by Wireshark to confirm that when we played the sound recording for the first time, a connection was made that delivered the audio data. We also confirmed that when we streamed the sound recording for the second time, there was no connection made to transfer audio data. From this, we were able to deduce that a copy of the sound recording must have been made and that this copy must have been stored to disk.

54. Having already identified the "radfile.rcf" file as the likely site of the copy, we then deleted that file. After that, we restarted the Wireshark tool to create a network trace, and streamed the same sound recording again. When we examined the new network trace captured by Wireshark, we were able to confirm that when we streamed the sound recording for this third time, a connection was made that transferred audio data just as if we had never played the sound recording before. Accordingly, the file "radfile.rcf" must have contained the copy of this sound recording. Our action of removing this file between the second and third instances of streaming the sound recording forced the Rhapsody player application to retrieve the audio data for this sound recording from the server again. If the copy of the audio data had been in some file other than "radfile.rcf," the service would not have needed to redeliver the audio data during the third instance of streaming it.

55. Overall, these experiments allow me to answer each of the questions posed in the Copyright Royalty Judges' March 11 Order with respect to the technology underlying the Rhapsody service. I set forth my answers below:

(a) **What technological process occurs when a listener selects a particular sound recording to be streamed from the perspective of the service offering the interactive streaming and any third party that facilitates that process?**

I have described the technological process of Rhapsody's interactive streaming service in detail above. In sum, when a subscriber uses the Rhapsody player application to select and stream a particular sound recording for the first time, the Rhapsody player establishes a TCP connection to a server machine for the purposes of transferring the encoded audio data. I have explained the general concept of using TCP for streaming in Section 4, and the process that occurs with Rhapsody specifically matches this general model. Our experimentation has further revealed that the Rhapsody service uses the Limelight CDN to provide this functionality. Finally, as a result of the Rhapsody streaming process, two copies of the audio data of the sound recording are made: (1) a copy in RAM on the user's computer that is used to play the sound recording; and (2) a cached copy stored on the hard drive of the user's computer to be used when the same sound recording is played again, rather than re-transferring audio data from the Rhapsody servers.

(b) **Is an audio file downloaded to the listener's computer?**

Yes. As I have just described, for sound recordings streamed through Rhapsody, data is transmitted to a user's computer, stored in RAM, and cached for future use and stored on the hard drive in a user's computer within the "radfile.rcf" file, which is located in the same directory where the Rhapsody player application is installed.

(c) **If an audio file is downloaded, is the listener able to access that file in any manner?**

Yes. The file "radfile.rcf" is accessible to the user like any other file on disk and can be copied and manipulated. Because this file is created and maintained by the Rhapsody player application to store all copies of audio data, some effort would be required for a user to extract the data for a specific sound recording from the file. In other words, exactly how data is organized and written to this file is proprietary to the Rhapsody player application, but through the application, an authenticated user is able to access the specific audio file.

(d) **If the listener is unable to access the downloaded file, what prevents such access?**

As noted above, the audio file can be accessed and played through the Rhapsody player.

(e) **How does the process for interactive streaming differ from that used for conditional or limited downloads?**

There is very little difference between the technological process for Rhapsody's interactive streaming service and the process for its conditional or limited download service.[14] Encoded audio data is transferred through a TCP connection with a server machine in both instances. And as described above, the results of each process are very similar in that choosing to stream a sound recording also leaves a cached copy of the

[14] For comparison purposes during our experiment, we also exercised the limited download features of each service. We observed that the audio data was transferred using TCP in a manner comparable to the transfer of data that is associated with the first time a song is played with interactive streaming. We were also able to observe that the limited download resulted in a copy of the file on the hard drive which was then used to play the song when selected in the player application. Again, this is comparable to how the cached copy of the song created during interactive streaming is used when the song is subsequently played after the first time.

sound recording on a listener's computer. Once a cached copy of the sound recording has been made on local storage in this way, the most significant difference between that and a sound recording stored as a limited download is simply where the copy of audio data is stored. For sound recordings transferred through interactive streaming, the copy is maintained within the file "radfile.rcf," while in the case of a conditional or limited download, sound recordings are stored as individual files on the user's computer. Other than this distinction, the process of making that copy, and the results of that process, are basically indistinguishable.[15]

5.7 MediaNet's Streaming Technology

56. MediaNet, formerly known as MusicNet, provides interactive streaming technology as well as other digital music technologies to third parties interested in offering subscription-based access to media resources. MediaNet's website states that its primary business is to offer technological infrastructure, including music content, to distributors who can then personalize their own individual direct-to-consumer music subscription services.

[15] Although both interactive streaming and limited or conditional downloading result in copies of an audio file being stored on the disk of the recipient's computer, in some instances, the copies may not be of the same quality. Copies transmitted as limited downloads may be encoded at a higher rate, which would result in a higher quality version of the sound recording. Nevertheless, complete copies of the audio data and sound recordings are made in both instances.

57. We performed our experiments using the Performer Digital service as an example of MediaNet's technology.[16] Performer Digital offers two different subscription levels. The basic subscription provides access to Performer Digital's entire catalog of music for interactive streaming and limited downloads. An enhanced subscription level allows the subscriber to copy limited downloads onto compatible portable music players. For our experiments, we subscribed at the basic subscription level.

58. Figures 4 and 5 show screenshots of the Performer Digital player application. The Performer Digital player application provides the user with tools for selecting music and creating and maintaining playlists, as shown in Figure 4. In Figure 5, a listing of sound recordings from a particular artist is shown. To play a particular sound recording, the user presses the play button next to the title.

---

[16] Performer Digital is MediaNet's own direct-to-consumer music service, and it appears to be the model MediaNet offers licensees who want a turnkey music subscription service. MediaNet is primarily operated as a business-to-business technology provider, and does not appear to market itself as a direct-to-consumer music provider. Performer Digital may, in fact, operate to model the various functions and options that can be provided by MediaNet technologies. Nevertheless, individual listeners can go to http://performerdigital.musicnet.com, subscribe to a music service and begin to download or stream music directly, as I did to examine MediaNet's technology for purposes of this report.

Figure 4

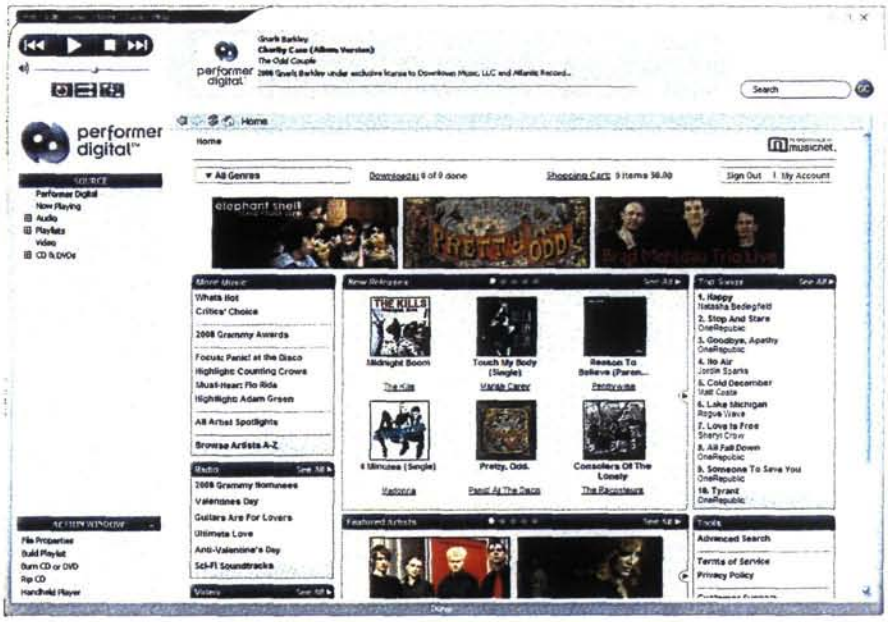
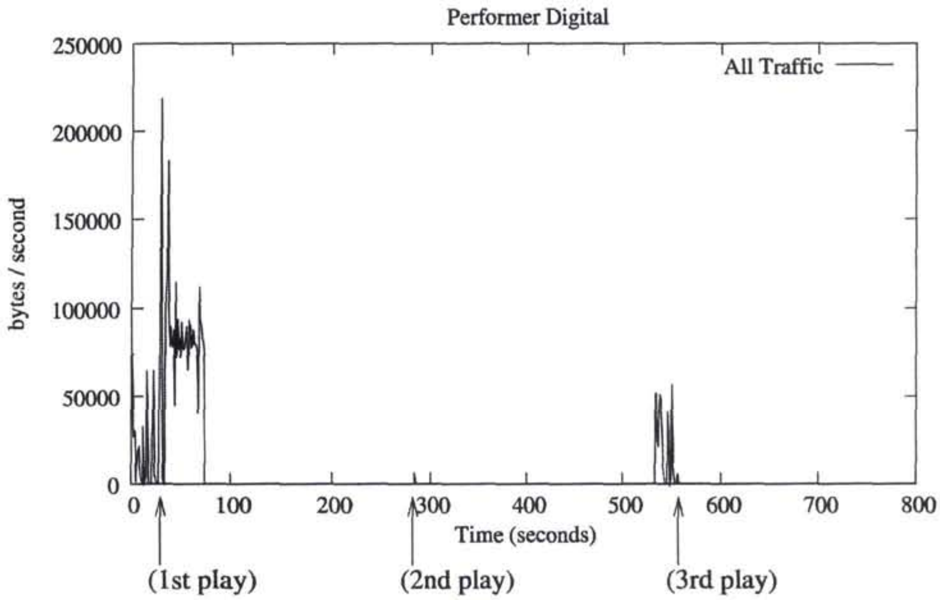


Figure 5

59. Exhibit D, constructed during the analysis procedure outlined above, shows the tabulated results from our experiment as applied to Performer Digital. During our experiment with Performer Digital, we used "Take Me Out" by the artist

Franz Ferdinand. The playing time for this sound recording is 3 minutes and 57 seconds. During the course of streaming this sound recording three times, a total of 68 TCP connections were made between the Performer Digital player application and the Performer Digital service. These connections communicated to 7 distinct server computers, all administered by MediaNet without the participation of third-party CDNs. Figure 6 shows the throughput of data sent to the player application over the course of the experiment.

Figure 6



60. As described at the beginning of this section, the data contained in this table can be analyzed to determine when audio data is transferred, and to deduce whether or not a copy of the sound recording has been made on the computer in addition to the RAM copy made for streaming. Our experiment and analysis revealed the following with respect to MediaNet's Performer Digital service:

(a) When the sound recording was played for the first time, the audio data was delivered by Connection Number 30. The sound recording we chose to stream was 237 seconds in length, and the number of bytes transferred by Connection 30, which began immediately after the sound recording was selected through the service, was 4,037,105 bytes. This represents a sound recording encoding rate of 170 kilobits/second for the length of the sound recording, which is within the range that we would expect. No other connection delivered an appropriate amount of data given the sound recording's length at an appropriate time in the experiment.

(b) Again, a copy of the sound recording was stored in the RAM of the computer we were using as a result of the data transfer process. In order to play the sound recording through the computer's speakers during the first playback, a copy of the audio data for that sound recording must have been buffered and stored in the RAM.

(c) A cached copy of the sound recording must also have been made, either on RAM or on disk in the computer's hard drive, since there was no connection during the second or third playback attempts that transferred an appropriate amount of data. The same analysis and logic used to come to this conclusion for the Rhapsody player application applies here. Graphically, this is seen by the absence of significant throughput activity in Figure 6 in the time immediately after the points marked "2nd play" and "3rd play."

(d) Using the same analysis we had applied to the Rhapsody service, it became clear that the cached copy of the sound recording was stored on disk. Briefly, because the Performer Digital application was exited and restarted between the second and third playback attempts, we can conclude that because the audio data was not transferred again during the third stream, the copy used to play the sound recording for the third time must have been stored on disk, rather than simply stored in RAM.

61. In fact, we were readily able to identify the copy of audio data made on the hard drive as a file within the folder "Temporary Internet Files." This location is used by a number of different Internet-related applications (most famously Internet Explorer) as temporary storage. Although the name of the file containing the audio data was a random combination of numbers and symbols – in this case "7457767_110.wma#0;1.000;0;0;1:2" – we were able to discern that the file contained the audio data because it was the appropriate size, was created at the time we originally streamed the sound recording, and contained the substring ".wma" as part of its name. ".wma" is a known identifier for Window Media Audio (WMA), an audio format developed by Microsoft.

62. Having located the file, we also discovered it was very easy to access and use it. We were able to copy it to another location and rename it. Furthermore, we were able to use the Window Media Player to play the file without having to start the Performer Digital application. This is not to say that the sound recording was totally unencrypted; it still appeared to be protected by DRM since we were unable to play it on a different computer which was not authorized as a MediaNet subscriber. Yet we believe that Performer Digital uses DRM technology from Microsoft, which is also built into the Windows Media Player.[17] This similarity explains why the Windows Media Player was able to properly authenticate our ability to play the sound recording in addition to the Performer Digital player. To confirm that the identified file

[17] MusicNet, *MusicNet Client SDK Guide: Programmer's Reference Guide 4.0.0.401*, April 2006, located at http://developer.musicnet.com/documentation/client.

was indeed a copy of the sound recording we played, we deleted the file and played the sound recording again and observed that audio data was transferred from the server. We were able to repeatedly observe that if the file was left alone, no audio data would be transferred, but if the file was deleted, audio data would be transferred the next time the sound recording was streamed.

63. In light of these results, I can now answer the questions raised by the Copyright Royalty Judges in their March 11 Order with respect to MediaNet's technology, as exemplified by the Performer Digital Service:

(a) **What technological process occurs when a listener selects a particular sound recording to be streamed from the perspective of the service offering the interactive streaming and any third party that facilitates that process?**

When a subscriber uses the Performer Digital player application to select and stream a particular sound recording for the first time, the Performer Digital player establishes a TCP connection to a server machine to transfer the encoded audio data. This interaction matches the model presented in Section 4. As a result of the streaming process used by Performer Digital, two copies of the audio data of the sound recording are made: (1) a copy in RAM on the user's computer that is used to play the sound recording; and (2) a cached copy stored on the hard drive of the user's computer to be used when the same sound recording is played again, rather than re-transferring audio data from the MediaNet servers.

(b) **Is an audio file downloaded to the listener's computer?**

Yes. As I have just described, for sound recordings streamed through MediaNet, data is transmitted to a user's computer, stored in RAM, and cached for future use and stored on the hard drive within a specific file located within the "Temporary Internet Files" folder on the computer.

(c) **If an audio file is downloaded, is the listener able to access that file in any manner?**

Yes. The file containing the copy of the sound recording stored within the "Temporary Internet Files" folder can be accessed and played through the Performer Digital application. Moreover, the file can be located, copied again to another location, renamed, and otherwise manipulated just like any other file on the listener's computer.

(d) **If the listener is unable to access the downloaded file, what prevents such access?**

As noted above, the file can be accessed and played through the Performer Digital application.

(e) **How does the process for interactive streaming differ from that used for conditional or limited downloads?**

As with Rhapsody, there is very little difference between the two technological processes or results. Audio data is transferred from the server to the listener's computer in the same way. And once a sound recording has been played, the copy of the sound recording maintained in the "Temporary Internet Files" folder acts exactly like a sound recording that has been provided and transferred as a conditional or limited download. Furthermore, because the copies are labeled individually, unlike with Rhapsody, the results of the two processes seem even more similar.

5.8 Napster's Streaming Technology

64. Napster offers two different subscription levels. The basic subscription level provides access to the entire Napster library of sound recordings from up to 3 different computers. The "Napster To Go" subscription level provides the

additional ability to copy sound recordings onto compatible portable music devices. In order to conduct our analysis, we subscribed to the basic Napster subscription.

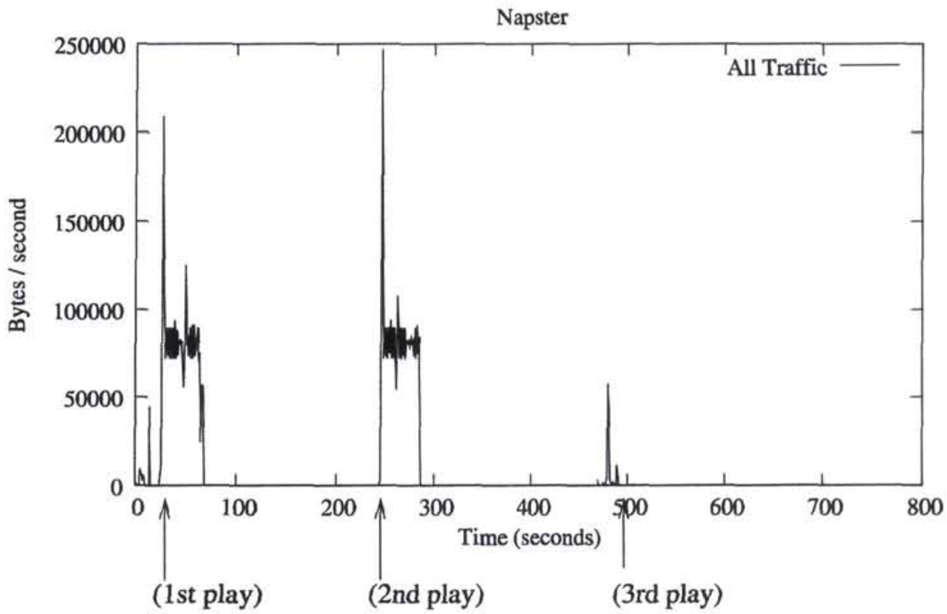
65. Figures 7 and 8 show screenshots of the Napster player application, which provides the listener with tools for selecting music and creating playlists. Some of these features can be seen in Figure 7. In Figure 8, a listing of sound recordings from a particular artist is shown. To play a particular sound recording, the user presses the play button next to the title.

Figure 7

Figure 8

66. Exhibit E shows the tabulated results from conducting our experiment with Napster. During this application of the experiment, we used "Float On" by the artist Modest Mouse. The playing time for this sound recording is 3 minutes and 28 seconds. A total of 52 TCP connections were made between the Napster player application and the Napster service as we streamed the sound recording three times. These connections communicated to 9 distinct server computers administered by three different business entities – Napster, STSN, and Akamai. Akamai is a well-known CDN provider and we believe STSN is also providing Napster with CDN functionality. Figure 9 shows the throughput received by the player application over the course of the experiment.

Figure 9



67. In regards to the Napster service, we learned the following through our experiment:

(a) When the sound recording was played for the first time, the audio data was delivered by Connection Number 24. As mentioned, the sound recording chosen to be played was 208 seconds in length. Connection 24, which began immediately after the sound recording was chosen to be played, was 3,537,284 bytes, representing a sound recording encoding rate of 170 kilobits/second for the length of the sound recording. This bitrate is in the range expected for audio data, and no other connection delivered an appropriate amount of data given the sound recording's length at an appropriate time in the experiment.

(b) As described with Rhapsody and MediaNet, in order for us to perceive the sound recording during the first playback, a copy of the audio data was necessarily stored in the computer's RAM. This copy was capable of being perceived, and was, in fact, perceived as the bits that comprised the audio file in RAM were sent, by the player application, to the computer's speakers.

(c) Unlike the Rhapsody and Performer Digital player applications, however, the Napster player application does not create a cached copy of the audio data after the first time a sound recording is streamed. During the second playback, data is delivered to the Napster player application just as it was during the first playback, creating another copy in RAM as well. Instead, if and when the sound recording is streamed a second time, a cached copy of the audio data will result. This can be seen graphically in Figure 9 in the absence of any significant throughput after the start of the third playback. In this experiment, we identified Connection 30 as the connection that delivered the audio data when the sound recording was streamed a second time. It is interesting to note that while the data was delivered from a server administered by Napster during the first playback, the second playback transferred data from an STSN server, which is apparently acting as a CDN for Napster. This pattern of receiving the audio data from Napster for the first play and then from STSN for the second play was a pattern that we were able to observe repeatedly by streaming other sound recordings.

(d) As we learned with Rhapsody and MediaNet, we discovered that the cached copy of the sound recording, made after the second playback in Napster's case, was stored on the computer's hard drive. Our exit from and reconnection to the Napster application between the second and third playback attempts confirms this, as a copy that was not stored on the disk would not have been available after exiting and restarting the application.

68. Upon further investigation, we were also able to determine where on the computer's disk Napster is storing cached copies of the sound recording. Like Performer Digital, these copies appear in the folder "Temporary Internet Files." The copy for "Float On," for example, was named "fe5eb5d4a31f30251058c81f04f182be.wma#0;1.000;0;0;1:2." Unlike the files created by Performer Digital, however, we were unable to also play the sound recording using the Windows Media Player, which likely means that these files are protected either by a different form of DRM or by a form of Windows DRM that is incompatible with the Windows Player.

69. Having presented the results of our experimentation, I can now answer the questions posed by the Copyright Royalty Judges in their March 11 Order with respect to Napster's technology:

(a) **What technological process occurs when a listener selects a particular sound recording to be streamed from the perspective of the service offering the interactive streaming and any third party that facilitates that process?**

When a subscriber uses the Napster player application to select and stream a particular sound recording for the first time, the Napster player establishes a TCP connection to a server machine for the purpose of transferring the encoded audio data. This interaction again matches the model presented in Section 4, and makes a copy of the audio data in RAM on the user's computer in order to play the sound recording. When a subscriber streams a particular sound recording a second time, audio data is again transferred, this time from an STSN server. In this instance, two copies of the audio data of the sound recording are made: (1) a copy in RAM on the user's computer that is used to play the sound recording; and (2) a cached copy stored on the hard drive of the user's computer to be used when the same sound recording is played again, rather than re-transferring audio data from the Napster servers.

(b) **Is an audio file downloaded to the listener's computer?**

Yes. As I have just described, for sound recordings played at least twice, the audio data is copied and stored in a specific file located in the "Temporary Internet Files" folder.

(c) **If an audio file is downloaded, is the listener able to access that file in any manner?**

Yes. The file containing the copy of the sound recording can be accessed and played by an authenticated user through the Napster player application, and can also be copied again to another location, renamed, and otherwise manipulated just like any other file on the listener's computer.

(d) **If the listener is unable to access the downloaded file, what prevents such access?**

As noted above, the file can be accessed and played through the Napster client.

(e) **How does the process for interactive streaming differ from that used for conditional or limited downloads?**

The primary difference is that a listener only needs to download a sound recording as a limited download once for a copy to be stored on the computer's hard drive. For sound recordings transferred through interactive streaming, the sound recording must be streamed twice before a cached copy is made on the user's computer. However, once a sound recording has been played twice, the copy of the sound recording maintained in the "Temporary Internet Files" folder acts exactly like a sound recording that has been provided as a conditional or limited download. Thus, there is very little difference between these two processes.

6. Conclusion

70. As demonstrated by my answers to the Copyright Royalty Judges' questions in the March 11 Order, my analysis of the participant services' interactive streaming technology, and my discussion of streaming technologies in general, I have reached a number of conclusions concerning the technological process of interactive streaming and the copies of sound recordings made as a result:

71. When an audio file is streamed using the network communication protocol used by MediaNet, Rhapsody and Napster – that is, TCP – a copy of the audio file is delivered to the player application program on the user's computer, and is stored in

buffer memory in RAM so that it can be played. In addition, each of the three services I examined deposits a cached copy of the streamed audio file on the user's computer's disk. MediaNet's Performer Digital service and RealNetworks' Rhapsody service create this cached copy the first time a sound recording is streamed; Napster's service creates this cached copy once a sound recording is streamed a second time. This additional copy of the audio file remains accessible indefinitely and can be reproduced, distributed and manipulated by the user after the audio file stream is completed. Therefore, based upon my investigation of MediaNet, Rhapsody and Napster, an audio file made available via interactive streaming is perceptible to a listener when the transmitted audio file of the sound recording is copied in RAM and played for the listener, *and* the audio file is also stored locally so that a copy of the sound recording is available to the listener for future playbacks without requiring the service to retransmit the file over the Internet.

72. My experiments also show that the processes used by these three services for interactive streaming do not differ significantly from the processes they use for conditional or limited downloads. There is little to no technological difference between the copies made during interactive streaming and those made as the limited or conditional downloads provided by these music services.

Declaration

I declare under penalty of perjury that the foregoing is true and correct.

Executed on: April 3, 2008

Ketan Mayer-Patel

Errata to the Expert Report of Ketan Mayer-Patel

The automatic section numbering of Section 5 begins at Section 5.5. It should, instead, begin at Section 5.1. Accordingly, the reference to Section 5.5 in paragraph 13 on page 7 should instead be a reference to Section 5.1.

The first sentence of paragraph 45 on page 22 reads: "If we did not observe a connection transferring a similarly appropriate amount of data when the sound recording was streamed a second time, we could deduce that a cached copy of the sound recording must have been made on the computer hard drive in addition to the RAM copy made as a result of the first stream of the sound recording."

That sentence should instead read: "If we did not observe a connection transferring a similarly appropriate amount of data when the sound recording was streamed a second time, we could deduce that a cached copy of the sound recording must have been made."

A

Ketan Mayer-Patel

311 Sitterson Hall
Department of Computer Science, CB #3175
University of North Carolina, Chapel Hill

kmp@cs.unc.edu
http://www.cs.unc.edu/~kmp

Education

Ph.D. University of California, Berkeley, 1999
Parallel Software-only Video Effects Processing

M.S. University of California, Berkeley, 1997
Design and Performance of the Berkeley Continuous Media Toolkit

B.A. University of California, Berkeley, 1992
Majors: Computer Science and Economics

Academic Positions

Associate Professor
University of North Carolina, Chapel Hill, NC. (August 2005 – present)

Assistant Professor
University of North Carolina, Chapel Hill, NC. (January 2000 – August 2005)

Visiting Researcher
Microsoft Bay Area Research Center (BARC), San Francisco, CA. (June 2003 – December 2003)

Graduate Student Researcher
University of California, Berkeley, CA. (June 1993 – November 1999)

Graduate Student Instructor
University of California, Berkeley, CA. (August 1997 – December 1997)

Programmer
University of California, Berkeley, CA. (June 1992 – June 1993)

Programmer
United States Department of Agriculture, Albany, CA. (May 1991 – June 1992)

Awards and Notables

National Science Foundation CAREER Award, 2003
Computer Science Student Association Teaching Award, 2003
In the fifteen-year history of the ACM SIGMultimedia Conference, considered to be the premier conference in the field of multimedia, I have published thirteen papers in ten different years.
Appointed to the editorial board of ACM Transactions on Multimedia Computing, Communications, and Applications, the flagship technical journal for multimedia.
Co-Chair of the standing executive committee for the International Workshop on Network and Operating System Support for Digital Audio and Video.

Current Research Areas

Coordinated Multistreaming
In this project, we are developing mechanisms to address the needs of distributed multimedia applications that employ many (i.e., 10's or 100's) of different media flows with complex

inter-stream semantics and adaptation requirements. This project addresses fundamental problems in protocol coordination and aggregate congestion control.

Multidimensional Adaptation
We are developing a framework for compactly expressing and evaluating adaptation policies that must negotiate tradeoffs in real-time within very large multiresolutional datasets with high dimensionality.

StrandCast
StrandCast is an application-layer multicast protocol intended for latency-insensitive multimedia applications such as receiver-driven layered multicast and pyramid broadcasting. The design and implementation of StrandCast exploits the lax latency requirements of these applications to optimize for link stress, rapid joins and leaves, and robustness in the face of node failure.

Scalable Display Architectures
This project addresses challenges faced when building truly scalable displays with 10's of millions or even billions of pixels. Existing frame buffer-based architectures do not scale to these sizes. As digital display technologies evolve, we can envision new display architectures that may place memory and computational capabilities at each pixel or groups of pixels. This project explores how such an architecture would be supported at the application level as well as by network protocols for supporting remote scalable displays.

Encoding and Transmission of 3D Scenes from Multiple Cameras
The project explores ways to efficiently transmit video data from a set of cameras viewing the same scene. This problem is at the heart of most tele-immersion applications. Our hypothesis is that it is possible to exploit depth information (even if imperfect) derived from stereo correlation between cameras to more efficiently encode the original color information.

Recoverable Video Adaptation
Existing video adaptation techniques generally lead to irreversibly loss of video quality. In this project, we are exploring adaptation techniques that can be used to recover high (or at least higher) quality video from a set of independently constructed lower quality representations.

<u>Select Publications</u>

K. Mayer-Patel, "Video Conferencing," chapter within *The Handbook of Computer Networks*, vol. 3. to be published by John Wiley & Sons, expected late 2007.

S. Krishnan and K. Mayer-Patel, "A Utility-Driven Framework for Loss and Encoding Aware Video Adaptation," *Proceedings of the 15th International ACM Conference on Multimedia*, Ausberg, Germany, 2007.

D. Ott and K. Mayer-Patel, "An open architecture for transport-level protocol coordination for distributed multimedia applications," to appear in *ACM Transactions on Multimedia Computing, Communications, and Applications*.

D. Gotz and K. Mayer-Patel, "GAL: A middleware library for multidimensional adaptation," to appear in *ACM Transactions on Multimedia Computing, Communications, and Applications.*

K. Mayer-Patel, B. Smith, and L.A. Rowe, "The Berkeley software MPEG-1 video decoder," *ACM Transactions on Multimedia Computing, Communications, and Applications,* vol. 1, no 1, January 2005.

D. Gotz and K. Mayer-Patel, "A general framework for multidimensional adaptation," *Proceedings of the 12th International ACM Conference on Multimedia,* New York, 2004, pp 612-619.

D. Ott and K. Mayer-Patel, "Coordinated multi-streaming for 3D tele-immersion," *Proceedings of the 12th International ACM Conference on Multimedia*, New York, NY, 2004, pp. 596-603.

D. Ott and K. Mayer-Patel, "Aggregate congestion control for distributed multimedia applications," *Proceedings of IEEE Infocom '04*, Hong Kong, 2004.

K. Mayer-Patel and W. Miaw, "Evaluating the effectiveness of automatic PVR management," *Proceedings of the SPIE Conference on Storage and Retrieval Methods and Applications for Multimedia*, San Jose, CA, January 2004.

S.-U. Kum, K. Mayer-Patel and H. Fuchs, "Real-time compression for dynamic 3D environments," *Proceedings of the 11th International ACM Conference on Multimedia*, Berkeley, CA, 2003, pp. 185-194.

K. Mayer-Patel, L. Le and G. Carle, "An MPEG performance model and its application to adaptive forward error correction," *Proceedings of the 10th International ACM Conference on Multimedia*, Juan-les-Prins, France, 2002, pp. 1-10.

D. Ott and K. Mayer-Patel, "A mechanism for TCP-friendly transport-level protocol coordination," *Proceedings of the USENIX Technical Conference*, Monterrey, CA, 2002.

D. Ott and K. Mayer-Patel, "Transport-level protocol coordination in cluster-to-cluster applications," *Proceedings of the 8th International Workshop on Interactive Distributed Multimedia Systems (Lecture Notes in Computer Science)*, vol. 2158, Springer, 2001, pp. 10-22.

K. Mayer-Patel, D. Simpson, D. Wu, and L.A. Rowe, "Synchronized continuous media playback through the World Wide Web," *Proceedings of the 4th International ACM Conference on Multimedia,* Boston, MA, 1997, pp. 435-436.

L.A. Rowe, K. Patel, and B. Smith, "MPEG video in software: representation, transmission, and playback," *Proceedings of the SPIE conference on Multimedia Computing and Networking*, vol. 2188, San Jose, CA, 1994.

K. Patel, B. Smith, and L.A. Rowe, "Performance of a software MPEG video decoder," *Proceedings of the 1st International ACM Conference on Multimedia,* Los Angeles, CA, 1993, pp. 75-82.

Select Software Artifacts

mpeg_play

The first publicly available MPEG-1 video decoder originally released in 1993. Over 1,000,000 copies of this program have been downloaded. It has been used as a code base for innumerable research and open source systems. Mayer-Patel was the architect of the original code that was later refactored and maintained by a number of other individuals.

The Berkeley Continuous Media Toolkit

The Berkeley CMT provided a framework within which to develop experimental multimedia tools and applications. Although primarily used by researchers at UC Berkeley, it was employed by a number of different research groups world-wide. Development of CMT ended in approximately 1998.

MPEG2Event

This recently released C# library allows researchers to rapidly develop MPEG-2 analysis tools that are interested in the details of bit-level coding elements. Although currently in use by only a small number of researchers, it is freely available at http://www.cs.unc.edu/~kmp/mpeg2event. Further development of the library is on-going.

Leadership and Professional Activities

Program Committees

- ACM Multimedia (2000, 2003, 2004, 2006, 2007)
- Workshop on Network and Operating System Support for Digital Audio and Video (2000, 2002, 2003, 2006, 2007)
- SPIE Conference on Multimedia Computing and Networking (2001, 2002, 2003, 2006, 2007)
- IFIP Networking Conference (2005, 2006, 2007)
- Multimedia Interactive Protocols and Systems Workshop (2004, 2005)
- Multimedia Information Systems Conference (2004)
- International World Wide Web Conference (2004)
- IEEE International Conference on Distributed Computing Systems (2003)
- Interactive Distributed Multimedia Systems Workshop (2003, 2002)
- Global Internet Symposium (2002, 2001)

Organizing Committees

- General Co-Chair, Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV (2005)
- Program Chair, Systems Track, ACM Multimedia (2006)
- Open Source Software Competition Chair, ACM Multimedia (2004, 2005)

- Tutorial Program Chair, ACM Multimedia (2003)
- Doctoral Symposium Chair, ACM Multimedia (2000, 2001)

Other Professional Service
- Associate Editor, ACM Transactions on Multimedia Computing, Communications, and Applications.
- Co-chair of the executive steering committee for NOSSDAV.
- Invited panelist for CoNEXT 2006, a European meeting of researchers in networking and multimedia.
- Guest Editor, Special Issue of Multimedia Systems Journal featuring expanded papers from the SPIE Conference on Multimedia Computing and Networking, 2003.
- In 2004, participated in a by-invitation meeting of leaders within ACM SIGMultimedia. A report of the meeting outlines important directions for multimedia research.
- Invited to and attended an international meeting of multimedia researchers being organized for Spring 2005 in Dagstuhl, Germany to discuss the future of multimedia research.

University Service
- Chair of Graduate Admissions, Department of Computer Science
- Faculty discussion leader for the Carolina Summer Reading Program

B

Exhibit B: Materials Reviewed

In preparing the attached report, I visited the following websites and reviewed the materials located there:

http://www.akamai.com

http://developer.musicnet.com/documentation/client/

http://www.gnuplot.info

http://www.limelight.com

www.napster.com

http://network-tools.com

http://www.pcmag.com/article2/0,2817,2200712,00.asp

http://www.pcmag.com/article2/0,2817,2209710,00.asp

www.performerdigital.com

http://performerdigital.musicnet.com

www.rhapsody.com

http://www.wireshark.org

http://www.xfig.org

The network trace files, as well as spreadsheets, plots, and figures derived from my analysis of those traces are contained in the CD that is attached as Exhibit F.

C

Exhibit C: Rhapsody Network Trace Data

Conn. Number	IP Address	Business Entity	Relative Start (sec)	Duration (sec)	Bytes Received	Average Data Rate (kbs)
1	207.188.0.31	Real Networks	5.85	7.02	93233	106.29
2	207.188.0.31	Real Networks	6.21	6.54	127819	156.34
3	207.188.0.31	Real Networks	12.74	100.82	118263	9.38
4	207.188.0.31	Real Networks	13.15	8.67	126420	116.64
5	207.188.0.31	Real Networks	22.36	246.55	64484	2.09
6	72.247.241.167	Akamai	25.67	70.62	11200	1.27
7	72.247.241.167	Akamai	25.67	70.62	18848	2.14
8	68.142.118.112	Limelight	38.69	6.79	653724	770.68
9	68.142.118.44	Limelight	45.81	35.04	2472067	564.42
10	207.188.0.31	Real Networks	272.07	150.63	18298	0.97
11	207.188.0.31	Real Networks	428.66	7.44	53496	57.53
12	207.188.0.31	Real Networks	429.09	6.99	162812	186.22
13	207.188.0.31	Real Networks	436.09	13.44	86355	51.39
14	207.188.0.31	Real Networks	436.37	13.18	168142	102.06
16	72.247.241.167	Akamai	448.15	61.81	1839	0.24
17	72.247.241.167	Akamai	448.16	71.08	1733	0.20

18	72.247.241.40	Akamai	449.38	60.52	860	0.11
19	72.247.241.40	Akamai	449.44	60.52	860	0.11
20	207.188.0.31	Real Networks	449.44	156.50	24333	1.24
21	207.188.0.31	Real Networks	449.44	69.71	5921	0.68

D

Exhibit D: MediaNet Trace Data

Conn. Number	IP Address	Business Entity	Relative Start (sec)	Duration (sec)	Bytes Received	Average Data Rate (kbs)
1	209.67.101.70	MediaNet	0.00	3.18	50569	127.05
2	209.67.101.70	MediaNet	0.20	3.37	91061	215.97
3	209.67.101.194	MediaNet	1.10	0.28	386	10.87
4	209.67.101.194	MediaNet	1.31	0.27	385	11.62
5	209.67.101.194	MediaNet	1.32	0.25	385	12.17
6	209.67.101.194	MediaNet	1.49	0.27	385	11.53
7	209.67.101.194	MediaNet	1.57	0.25	385	12.20
8	209.67.101.194	MediaNet	1.66	0.27	385	11.46
9	209.67.101.194	MediaNet	1.74	0.25	385	12.10
10	209.67.101.192	MediaNet	1.76	0.25	387	12.49
11	209.67.101.192	MediaNet	1.88	0.26	386	11.95
12	209.67.101.192	MediaNet	1.92	0.27	386	11.60
13	209.67.103.4	MediaNet	2.32	0.69	4512	52.44
14	209.67.103.4	MediaNet	2.32	0.63	6079	77.80
15	209.67.103.4	MediaNet	2.86	0.25	1816	59.31
16	209.67.103.4	MediaNet	2.93	0.24	4272	139.57
17	209.67.101.70	MediaNet	3.10	75.24	109305	11.62
18	209.67.101.70	MediaNet	3.48	12.59	52928	33.62
19	209.67.101.173	MediaNet	14.36	0.51	915	14.25
20	209.67.101.172	MediaNet	14.84	0.75	4097	43.79
21	209.67.101.172	MediaNet	15.54	0.49	1377	22.38

22	209.67.101.70	MediaNet	15.99	72.35	123532	13.66
23	209.67.101.172	MediaNet	16.07	0.66	2488	30.19
24	209.67.101.173	MediaNet	21.78	0.33	912	21.84
25	209.67.101.172	MediaNet	22.04	1.27	61665	388.30
26	209.67.101.172	MediaNet	23.18	0.34	1420	33.38
27	209.67.101.172	MediaNet	23.45	0.53	14181	214.19
28	209.67.101.172	MediaNet	27.74	0.48	1320	21.99
29	209.67.103.4	MediaNet	28.34	19.86	6178	2.49
30	209.67.102.103	MediaNet	28.37	46.41	4037105	695.97
31	209.67.102.103	MediaNet	74.71	0.34	598	14.14
32	209.67.102.103	MediaNet	251.56	0.35	628	14.23
33	209.67.101.172	MediaNet	282.93	0.47	1260	21.46
34	209.67.103.4	MediaNet	283.42	20.78	6178	2.38
35	209.67.102.103	MediaNet	504.18	0.34	628	14.61
36	209.67.101.70	MediaNet	532.90	3.96	12707	25.70
37	209.67.101.70	MediaNet	533.10	3.36	119172	283.57
38	209.67.101.194	MediaNet	534.03	0.27	386	11.43
39	209.67.101.194	MediaNet	534.26	0.26	385	11.70
40	209.67.101.194	MediaNet	534.29	0.26	385	11.93
41	209.67.101.194	MediaNet	534.43	0.26	385	11.82
42	209.67.101.194	MediaNet	534.55	0.27	385	11.39
43	209.67.101.194	MediaNet	534.61	0.26	385	11.68
44	209.67.101.194	MediaNet	534.73	0.27	385	11.52
45	209.67.101.192	MediaNet	534.76	0.25	387	12.18

46	209.67.101.192	MediaNet	534.81	0.26	386	11.90
47	209.67.101.192	MediaNet	534.94	1.31	386	2.35
48	209.67.103.4	MediaNet	535.39	0.68	4512	53.22
49	209.67.103.4	MediaNet	535.39	0.65	6079	75.19
50	209.67.103.4	MediaNet	535.93	0.45	1816	32.30
51	209.67.103.4	MediaNet	535.99	0.42	4338	83.14
52	209.67.103.4	MediaNet	536.32	9.08	198	0.17
53	209.67.101.70	MediaNet	536.38	71.64	104344	11.65
54	209.67.101.70	MediaNet	536.77	13.65	134149	78.60
55	209.67.101.173	MediaNet	547.85	1.51	915	4.83
56	209.67.101.172	MediaNet	549.10	0.90	4097	36.62
57	209.67.101.172	MediaNet	549.95	0.43	1437	26.62
58	209.67.101.70	MediaNet	550.34	67.68	64836	7.66
59	209.67.101.172	MediaNet	550.40	0.60	2488	33.41
60	209.67.101.172	MediaNet	555.02	0.45	1260	22.28
61	209.67.103.4	MediaNet	555.96	16.43	6118	2.98
62	209.67.101.192	MediaNet	556.68	0.25	386	12.39
63	209.67.101.192	MediaNet	562.27	0.25	386	12.34
64	209.67.101.192	MediaNet	567.29	0.25	386	12.49
65	209.67.101.192	MediaNet	572.21	0.27	386	11.42
66	209.67.101.192	MediaNet	577.79	0.26	386	12.09
67	209.67.101.192	MediaNet	582.85	0.26	387	11.80
68	209.67.102.103	MediaNet	776.43	0.33	628	15.31

E

Exhibit E: Napster Network Trace Data

Conn. Number	IP Address	Business Entity	Relative Start (sec)	Duration (sec)	Bytes Received	Average Data Rate (kbs)
1	63.241.55.183	Napster	0.00	65.58	4927	0.60
2	63.241.55.183	Napster	4.09	1.15	5284	36.60
3	63.241.55.183	Napster	5.17	0.42	2169	41.66
4	128.109.34.45	Akamai	5.19	75.40	7575	0.80
5	128.109.34.45	Akamai	5.19	70.39	45809	5.21
6	63.241.55.183	Napster	5.44	70.14	7276	0.83
7	63.241.55.183	Napster	5.44	70.14	10020	1.14
8	63.241.55.183	Napster	5.51	0.80	2161	21.68
9	63.241.55.183	Napster	5.92	84.66	3079	0.29
10	63.241.55.183	Napster	6.23	0.37	861	18.72
11	63.241.55.183	Napster	6.32	0.28	898	25.78
12	63.241.55.105	Napster	6.36	0.48	2098	34.96
13	63.241.55.183	Napster	6.52	0.30	894	23.90
14	128.242.125.9	Akamai	6.66	17.40	2481	1.14
15	63.241.55.183	Napster	6.74	7.37	842	0.91
16	128.109.34.38	Akamai	8.36	67.22	1147	0.14
17	128.242.125.9	Akamai	24.46	12.40	907	0.59
18	63.241.55.183	Napster	24.47	0.38	1178	24.57
19	63.241.55.183	Napster	24.78	0.64	1456	18.34
20	63.241.55.183	Napster	24.79	0.74	1458	15.81
21	199.106.95.240	STSN	25.56	0.06	665	90.30

22	199.106.95.240	STSN	25.67	0.10	658	55.03
23	63.241.55.192	Napster	25.77	0.27	723	21.09
24	63.241.45.59	Napster	25.96	218.86	3537284	129.30
25	128.242.125.9	Akamai	244.33	10.16	907	0.71
26	63.241.55.183	Napster	244.33	0.99	2513	20.31
27	63.241.55.183	Napster	244.81	0.48	1517	25.24
28	199.106.95.240	STSN	245.42	0.06	665	85.13
29	199.106.95.240	STSN	245.52	0.08	725	68.39
30	199.106.95.209	STSN	245.58	225.52	3544773	125.75
31	63.241.55.183	Napster	306.11	5.33	886	1.33
32	128.109.34.45	Akamai	464.78	0.42	423	8.10
33	63.241.55.183	Napster	470.20	2.01	2180	8.66
34	63.241.55.183	Napster	475.18	64.68	4927	0.61
35	63.241.55.183	Napster	478.87	0.86	5306	49.13
36	63.241.55.183	Napster	479.94	1.70	4236	19.98
37	128.109.34.45	Akamai	479.96	69.90	56615	6.48
38	128.109.34.45	Akamai	480.02	99.84	4059	0.33
39	63.241.55.183	Napster	480.47	69.38	12298	1.42
40	63.241.55.183	Napster	480.47	69.38	21739	2.51
41	63.241.55.183	Napster	481.02	68.84	7408	0.86
42	63.241.55.183	Napster	481.02	68.83	708	0.08
43	63.241.55.183	Napster	481.56	0.31	862	22.19
44	63.241.55.105	Napster	481.72	0:56	2098	30.16
45	63.241.55.183	Napster	481.79	3.31	898	2.17

46	128.242.125.9	Akamai	481.98	9.33	907	0.78
47	63.241.55.183	Napster	485.03	0.30	898	23.84
48	63.241.55.183	Napster	485.25	64.60	2118	0.26
49	199.106.95.240	STSN	488.66	0.08	605	57.74
50	199.106.95.240	STSN	488.79	0.10	725	57.94
51	199.106.95.209	STSN	488.87	212.11	12503	0.47
52	63.241.55.183	Napster	700.26	1.83	2136	9.36

F

Exhibit F to the

Witness Statement of Ketan Mayer-Patel

Submitted with the

Copyright Owners Disk Exhibits