Internet Music:
Dream or (virtual) Reality?

FABIO KON
f-kon@cs.uiuc.edu
Department of Computer Science
University of Illinois at Urbana-Champaign
1304 W. Springfield Avenue.
Urbana, IL 61801 — USA

FERNANDO IAZZETTA
iazzetta@exatas.pucsp.br
Laboratório de Linguagens Sonoras
Comunicação e Semiótica — PUC-SP
R. Ministro Godoy, 969 - sala 4B-06
São Paulo - SP 05015-901 — Brasil

Abstract

The recent explosive growth of the Internet and the fact that personal computers with multimedia capabilities are now a commonplace have raised the interest in distributed multimedia systems. Among all multimedia applications, the ones that relate to interactive music are the ones which present the tightest timing constraints. Traditional operating systems and networking infrastructure do not provide enough support for the quality of service these applications require.

This paper describes a number of problems concerning distributed musical applications on the Internet and discusses possible solutions for each of them. We, then, consider a variety of situations in which Internet Music systems could be used and suggest that the Internet is a viable environment for musical activities even when using existing technologies.

1 Introduction

In 1995, our research group at the University of Illinois developed *Vosaic*, an extension to the Mosaic Web Browser that is capable of streaming compressed video and audio in real time over the Internet using adaptive algorithms [Chen et al., 1996, Tan et al., 1996]. This pioneering work demonstrated that it was possible to send complex multimedia data over the Internet in real-time. However, it also showed that the complete unpredictability of the Internet behavior would pose very significant obstacles to the development of interactive distributed applications. At the present moment, the Internet is characterized by large delays on message transmission and by large variations in these delays. These characteristics are known as *latency* and *jitter*, respectively.

The difficulties in dealing with multimedia data on the Internet are intensified when this data represents music. This happens because music presents much harder bandwidth and timing constraints than other kinds of applications. Currently, *Internet Music* is almost completely limited to one-way streaming of digitized audio with very little interaction. This paper discusses the viability of using the Internet for distributed musical applications addressing both artistic and technical concerns.

1.1 Musical Challenges

Music is a temporal art and each culture or each period in its history has managed to develop appropriate procedures to deal with timing matters in music. These procedures were determined by contextual constraints such as the development of musical instruments, the space where music was presented, the ability of performers in controlling the sound production with their instruments, and by the development of musical language itself.

For example, during Middle Ages, the acoustic resonance of large cathedrals where the Gregorian chant usually took place imposed some tempo constraints: a monophonic chain of long notes was necessary to guarantee the comprehension of the text that was sung. This situation was completely different during the Baroque and Classical periods when a more favorable acoustic, the technological development of musical instruments, and a sophisticated system of notations allowed a very refined manipulation of sound in relation to time.

In the current decade, the Internet is becoming a new space for the realization of music and, for this reason, it is necessary to develop new tools and procedures to handle the peculiarities of the Internet medium. These developments will only be possible with work by researchers in the music and computer science communities.

1.2 Technological Challenges

Multimedia data is not only voluminous but also highly dependent on timing constraints. When a server sends a 256kbps CD-quality audio stream through an overseas connection, it is essential that the network be capable of sustaining an average data rate of at least 256kbps. But this is far from being enough. If the audio is being played in real-time in the client side, it is also important that the network jitter be small. In other words, the data must arrive at that rate all the time; it cannot stop for a fraction of a second and resume later.

The effects of network jitter can be minimized by buffering the data in the client side and delaying the start of the playback for some seconds. This is the technique adopted by all existing audio streaming software. Although it works well for some kinds of applications, it is intolerable for a large number of applications which requires a higher degree of interactivity.

Music performance is an extreme case of sensitivity to timing constraints during interaction. In applications such as video conference and collaborative work, a delay of up to one or two seconds is tolerable. However, in music, if two performers are playing together, a delay in the order of hundredths of a second is noticeable. In some cases, a delay of a tenth of a second might be fatal.

We have performed an informal experiment in the *Laboratório de Linguagens Sonoras* at PUC/SP using a MAX patch to delay the sound produced by one of the musicians playing a duet. We noticed that, after some practice, the musicians were able to compensate for delays of up to 70 or 80ms and still play together. Larger delays started to be intolerable.

The current Internet and Operating System infrastructures are not suitable for distributed music applications. However, a number of well-known computer science techniques developed in this decade can be used for making the dream of Internet Music come true. We believe that this dream not only can come true but that it will be true in the first decade of the next millennium.

Section 2 presents two examples of current work in Internet Music. Section 3 describes the technological problems that Internet Music faces and discusses possible solutions. Section 4 presents a variety of musical applications on the Internet. We present our conclusions in section 5.

2 Related Work

During the last decade, very little research has been carried out in the field of Internet Music. However, the recent explosion in the growth of the Internet and the fact that inexpensive personal computers are now capable of offering high-quality multimedia make this scenario likely to change.

NetSound [Casey and Smaragdis, 1996] is a sound and music description system that is capable of streaming *Csound* [Vercoe, 1997] specifications in real-time. The main advantage of this approach is an enormous economy in network bandwidth. Instead of sending a digitized audio wave through the network, *NetSound* sends a mathematical description of the sound wave which is then generated at the target machine. Therefore, instead of requiring bandwidth in the order of kbps or Mbps, it transmits data at rates in the order of hundreds of bits per second. The powerful *Csound* engine is used to synthesize high-quality audio in real-time at the target machine.

Although this approach solves the bandwidth problem, it does not address latency and jitter issues (see section 3.3). Actually, it might increase these problems since intensive computation is performed in the target machine.

The *Global Visual-Music Project* [Puckette et al., 1998] is developing an infrastructure for real-time, networked multimedia for multi-site, improvisatory performance. This group has developed PD, a programming language for real-time audio and graphics applications. By using this language and a set of extensions called GEM [Danks, 1997], the group has composed *Lemma 1* for percussion and trombone. It was performed on September 1997 in a Jazz club during the International Computer Music Conference. This performance was the first phase of a series called *Global Visual-Music Jam Session.*

The second phase includes *Lemma Two* in which performers in two different locations will interact through networked digital equipment. Its first performance is scheduled for Fall 1998.

In order minimize quality of service problems with latency and throughput, the communication is done through dedicated ISDN connections. This approach makes its cost prohibitive for everyday use and for the vast majority of musicians and computer music researchers. A more generic solution is required.

These two research efforts bring important contributions to the future of Internet Music but there are still a large number of problems that must be addressed by future systems. We now describe each of these problems.

3 Technological Infrastructure

Most multimedia applications have in common the fact that they require real-time manipulation of large quantities of data. In the case of distributed multimedia, this data must be processed in, at least, two different machines and transmitted over a network.

Users dealing with applications such as video-on-demand and multimedia presentation systems can tolerate delays in the order of some seconds. There is no problem on waiting five seconds to watch a two-hour movie. When it comes to audio or videoconferencing tools, a delay of up to one second can be acceptable. If the delay is higher than that, the conversation will not flow smoothly.

Distributed multimedia system developers have experienced that existing operating system and networking infrastructures are not tailored to support the real-time guarantees that these applications require. The protocols on which the Internet is based were designed for exchanging textual information and for transferring files. They are not optimized for large amounts of data, for meeting deadlines, and for dealing with heterogeneous environments.

As mentioned before, Internet Music applications are the multimedia applications that are most sensitive to timing constraints. In this section, we discuss how multimedia research has been trying to solve these technological problems and how their approaches can be applied to the challenge of making Internet Music a reality.

3.1 Data Compression

The first problem one faces when dealing with professional-quality music in a digital environment is the fact that audio data is voluminous. Thus, one can expect the need for large storage areas, high-bandwidth connections, and high processing power. Luckily, this is not only a problem for music. In general, video applications require much more resources than audio.

CD-quality audio requires two 16-bit tracks containing 44.1 thousand samples per second, which is equivalent to a data rate of 1411.2kbps. Recent research suggests that a higher sample rate and a larger sample size could still improve the human-perceptible sound quality. But this data rate already leads to a storage requirement of approximately 635Mbytes per hour.

This problem can be solved by applying data compression techniques. However, traditional compression mechanisms such as the Lempel-Ziv algorithm – implemented by programs like `gzip` – generally does not achieve a good compression rate when applied to audio data.

Audio-specific compression algorithms have been developed as a means to utilize the knowledge about the characteristics of the audio data in order to achieve a better compression ratio. The GSM algorithm [Vary and et al, 1988], for example, is used in audioconferencing and telephony applications and produces a data rate of 11.3kbps. But its quality is far from being enough for professional musical applications.

The MPEG layer 3 standard [MPEG.ORG, 1998] uses a psychoacoustic model to capture the information that is more relevant to the human ear and ignores what cannot be perceived [Sporer et al., 1992]. By applying filter banks, quantizations, entropy compression, and by exploiting redundancy of both channels, MPEG is capable of encoding high-quality audio at data rates ranging from 8kbps to 128kbps[1]. Near-CD quality can be achieved at 128kbps, resulting in a compression ratio of more than ten times.

The main problems that compression of high-resolution audio brings are the requirement for extra computational power and the delay that it imposes. Theoretically, MPEG layer 3 adds a delay of, at least, 59ms. The actual values, however, tend to be much higher than that depending on the implementation. Typically, existing hardware implementations add a delay of 150ms. Real-time encoding in software is not yet possible.

3.2 Alternative Data Representations

As we saw, data compression adds a significant delay and is not always possible in software. Besides, high-quality compressed audio still yields large amounts of data and data rates in the order of 128kbps. These problems can be avoided by using alternative data representations rather than dealing with the digitized wave form.

The most commonly known alternative representation is the MIDI format that stores information about musical notes rather than sound waves [Loy, 1985]. The MIDI standard produces very low data rates typically in the order of 0.1 to 5kbps. The codified notes are received by the MIDI player which produces the output sound wave. This can be done either by manipulating wave tables or by synthesizing the wave forms from mathematical models. This process can be done both in software and using special hardware such as sound cards or other sound devices.

MIDI presents a series of limitations [Moore, 1987, Puckette, 1994] including the fact that the data stream does not contain a description of the wave form but, simply, pitch, intensity, and the "instrument" that plays each note. Each MIDI player is free to give its own interpretation of how the instrument will sound.

A better solution would be to extend the representation to include detailed information about the instrument's timbre. This approach is taken by the MOD format, first used in Amiga computers, which includes samples of the instruments' wave forms. However, MOD does not provide mechanisms for fine grain manipulation of the wave forms once they are transmitted. Total control of the sound wave is achieved in NetSound (see section 2) which transmits Csound specifications that are synthesized in real-time in the target machine.

We can clearly notice a significant trade-off in all these approaches. The higher is the control over timbre and wave form, the higher is the computation power requirement and processing delays. MIDI requires very little resources, incurs very little delays but does not provide control over the wave form. NetSound requires high-performance CPUs, imposes higher delays but enables the tightest control.

[1] The higher the data rate is, the higher is the sound quality.

3.3 Networking Support

Research in networking support for real-time applications has been trying to solve three major problems: bandwidth, latency, and jitter. The **bandwidth** problem is the one that has received most attention from the industry and government. The bandwidth of existing wide-area networks has been increasing very rapidly in the last few years. According to Richard Palmer, a marketing director at Cisco Systems, Inc., new gigabit switch routers will scale from 622Mbps to 2.4Gbps in 1998 and to 9.6Gbps in 1999. *Internet2* will connect more than one hundred research universities in the United States at 2.4Gbps by the year 2000 [University Corporation for Advanced Internet Development, 1998]. The same trend can be observed in other countries and in non-academic Internet systems.

These high-speed connections will enable the transmission of high-quality audio in real-time to a large number of sites. However, it does not solve the problems of latency and jitter.

In Internet Music, **latency** can be defined as the time between a musical event happens on one side of the Internet connection and the time in which it is reproduced on the other side. **Jitter** is the variance in the latency. Consider, for example, a MIDI keyboard that sends its output to a remote site over an Internet connection. On the other side of this connection, a synthesizer produces the sound corresponding to the notes sent through the connection. Assume a musician plays a C in the keyboard every second. The time elapsed from the instant in which the musician presses the C key and the instant in which the C is played on the other side is the latency l. But, l is not always the same. Depending upon the network and operating system state, hardware, and software, l can vary a lot from one note to the following one. This variance, which produces disastrous effects in musical applications, is called jitter.

Latency can be minimized and jitter can be almost completely eliminated by deploying new networking protocols and hardware. Commonly used Ethernet connections and the Internet Protocol does not provide any kind of guarantee regarding the transmission of data packets. One cannot know, a priori, what the available bandwidth is and the latency can vary from a few microseconds to several seconds.

ATM and FDDI networks [Steinmetz and Nahrstedt, 1995], on the other hand, provide the basis for resource reservation and admission control. On these networks it is possible to reserve a certain fraction of the network time to specific data streams. Therefore, the network provides the user some guarantees regarding bandwidth, latency, and jitter.

A new and much more powerful version of the Internet Protocol, called IPv6, provides the underlying support for implementing systems capable of streaming real-time data guaranteeing that bandwidth, latency, and jitter requirements are met [Deering and Hinden, 1997]. A large number of operating system vendors and research groups are currently implementing IPv6 on their systems. It promises to be the next standard for wide-area inter-networking.

Once IPv6 is combined with fast, real-time-capable networks such as ATM, it will be possible to rely on the network to deliver multimedia packets with a very low latency and jitter. We believe that in the first decade of the new millennium, personal computers will be able to connect distant musical spaces – within a country – with imperceptible latency and jitter. In fact, network quality of service is a key component of the research supported by the Next Generation Internet and Internet2 initiatives.

It is important to point out that our planet is large enough so that the network delay between certain points in its surface will always be perceptible. The distance between São Paulo and Tokyo – when traveling on the Earth surface – is 18,530km. Thus, traveling at the speed of light, a data packet will take no less than 61ms to go from one site to the other. This value is very close to the human perception limit but current technology is very far from providing data transport with this kind of minimal delay. Therefore, it is likely that musicians playing together in these two locations will always perceive some delay between them.

3.4 Operating System Support

Existing commercial operating systems such as MS-Windows implement technology developed 25 years ago and, therefore, are suitable for meeting the requirements of applications used a quarter of a century ago. Multimedia and real-time applications have completely different requirements that have been addressed by recent research [Steinmetz and Nahrstedt, 1995].

Traditional operating systems manage hardware resources such as CPU, disk, memory, and network using best-effort strategies. They usually accept any number of tasks without paying attention to the quality of service that will be offered to their users. If one starts two video sessions and the compilation of a large program, the system will divide the CPU among all these tasks without taking into consideration the application-specific requirements. The video quality will probably be degraded even if there were enough resources for properly executing all tasks.

Real-time operating systems use special algorithms to support applications which must meet specific deadlines and receive a guaranteed portion of the machine resources. A video-on-demand application, for example, must be able to receive a certain amount of bytes per second from the network, receive a certain portion of the CPU time to decode the data stream and be able to send data out to be displayed on the screen and played on the speaker.

Experimental systems such as Nemesis [Leslie et al., 1996] controls the admission of new tasks and provide support for resource allocation and real-time scheduling based on application requirements. With these facilities, a system can offer guarantees that the multimedia streams will be processed with the required quality of service. Thus, the development of Internet Music applications are greatly facilitated on such environments.

4 Internet Music Applications

There are endless possibilities for applying Internet technology to music. In this section, we discuss some scenarios in which the topics discussed previously can be deployed.

4.1 Concert Broadcasts

At the University of Illinois, we have been developing a scalable distribution framework for real-time multimedia streaming [Kon et al., 1998]. Using this framework, we were able to build large distribution networks that could potentially serve 3,000 simultaneous clients with low-bandwidth video and audio.

Our technology was chosen to broadcast, live over the Internet, the coverage of the NASA JPL Mars Pathfinder mission [Golombek et al., 1997]. A network of more than 30 *Reflectors* spread across five continents was able to deliver live video and audio from the NASA Jet Propulsion Laboratory to more than one million clients in dozens of different countries.

This distribution framework is also capable of supporting the broadcast of CD-quality audio of a live concert or a MIDI-like stream. The only limitation is the available network bandwidth on the client Internet connections and on the Reflector sites.

Since this system currently runs on top of traditional operating systems and use the existing networking infrastructure, there are no guarantees in terms of bandwidth, delays, and jitter. We minimize this problem by buffering some data in the client before displaying it. This introduces additional delays that range from some tenths of a second to a couple of seconds.

Therefore, we have demonstrated that with existing technology it is possible to use the Internet for large-scale multimedia broadcasts and even for videoconferencing.

These delays are still too big for interactive music performances. But, since they are acceptable for videoconferencing applications, it would also be acceptable for master classes, workshops, and other forms of distant music learning.

4.2 Distributed Rehearsals and Concerts

Once the next generation of internet protocols are deployed and gigabit networks supporting quality of service are common, it will be possible to carry out distributed music performances. This will first be possible using dedicated links and, later, on the public Internet.

As happens with videoconferencing tools, each room participating in a distributed music session will be equipped with a conferencing system. This system will be responsible for (1) capturing the musical data generated on that room and sending it to the other participants, listeners, or to a Reflector, and (2) receiving musical data from remote sites and playing it locally.

Such a system will enable both distributed rehearsals and concerts. A conventional group, like a string quartet, would be able to "meet" several times a week to rehearse even if the group members live in different countries. The computer system will guarantee that the networking latency will not produce human-perceptible delays. High-quality audio and large-screen video will provide musicians with a rehearsing environment similar to the traditional one. If required, a limited number of "traditional rehearsals" could take place in the days preceding the concert.

However, even the concert could take the form of a distributed event with groups of musicians and audience, all in different locations. In that way, it will become much easier for musicians from different places to interact with each other and exchange musical experiences and knowledge.

4.3 Remote Supercomputing Resources

Supercomputing brings us the opportunity of experimenting with tomorrow's computing resources. A supercomputer today has the computational power of personal computers of five to ten years in the future. Musical applications of supercomputing include real-time synthesis [Kriese and Tipei, 1992], and computationally intensive algorithmic composition tools such as *MaxAnnealing* [Iazzetta and Kon, 1995].

Supercomputers are, by definition, scarce resources to which very few people have access. They are usually shared by a large number of researchers and they have very little mobility. Therefore, it is very unlikely that a supercomputer could be taken to a concert hall in order to perform music in real-time.

However, proper network support would enable the real-time connection of a supercomputer to other computers in a concert hall. Musicians could interact with local devices that would forward information to remote locations where the supercomputing power is available. Upon completion of the computation (e.g., sound synthesis, or algorithmic composition of parts of a piece) the result would return to the concert location and local systems would turn the data into music.

4.4 Composition of Internet Music

While final solutions for the technological problems are not found, we can think of a number of issues that contemporary composers should keep in mind when composing music for the Internet.

As stated in section 1.1, the design of interactive music systems on the Internet must take into account the peculiarities of the specific types of music they are intended for. What may be seen as extremely relevant in a particular context may not be pertinent to a different musical situation. There are at least three timing factors that will guide the design of an Internet music system.

1. the existence of a regular pulse: pieces based on regular timing structures (pulse, beats, bars) are much more sensitive to delays and jitters. Certain kinds of improvisational performances would be less affected by these constraints;

2. tempo: the tolerance on timing deviations is inversely proportional to how fast a piece is played and to the rate at which the events occur. Thus, a 100ms delay can be irrelevant for a slow piece but very disturbing for a very fast one.

3. compositional style: in a free improvisation – as the ones found in some of George Lewis and John Cage's pieces, for example – the performer can build musical structures as he

receives the sound data from the Internet. On the other hand, in pre-composed music, musicians on both sides of an Internet connection must be listening to the same information at the same time, or with a very short delay.

It is known that a trained musician is able to compensate for small timing deviations, as in the case of a timpanist who plays slightly ahead of the string session in a symphonic orchestra to compensate the distance the sound must travel from the back of the stage to the audience. But the delay must be very small and constant. Since there will always be a delay in the transmission of information over the Internet the issue that must be addressed here is the following. For a specific kind of music, how small the delay must be to provide a satisfactory level of interaction between different players.

By keeping in mind the specific characteristics of this medium, an Internet Music Composer can compensate for the current technological limitations. At the current stage of the Internet development, having a good knowledge about existing technical limitations is fundamental for the composition of Internet Music. As technology advances and the problems described in this paper are addressed and solved, it will become increasingly easier for composers to utilize this new medium.

5 Conclusions

Within the computer music community there is an increasing interest in the Internet in general and in Internet Music in particular. In spite of that, very few research groups and composers are working with distributed computational systems for music performance, composition, or education.

We have demonstrated that, although the existing computational infrastructure still presents significant problems, they all can be solved. Recent research in the areas of real-time systems, quality of service for multimedia, networking, and distribution mechanisms assure that a sound set of mechanisms can be combined to produce an ideal environment for distributed music applications on the Internet.

Composers can expect to have access to these environments in the first decade of the next millennium. But, one must not wait for them passively as there are important aesthetic questions regarding the use of the Internet for music practice. These questions are to be posed by the artistic community and answered by research in art, science, and technology.

References

[Casey and Smaragdis, 1996] Casey, M. and Smaragdis, P. (1996). NetSound. In *Proceedings of the 1996 International Computer Music Conference*, Hong Kong.

[Chen et al., 1996] Chen, Z., Tan, S. M., Sane, A., Li, Y., Campbell, R., and Xie, D. (1996). Video and Audio: Organization and Retrieval in the WWW. White Paper. Available at `http://vosaic.com/corp/papers/www5.html`.

[Danks, 1997] Danks, M. (1997). Real-time Image and Video Processing in GEM. In *Proceedings of the 1997 International Computer Music Conference*, pages 220–223, Thessaloniki.

[Deering and Hinden, 1997] Deering, S. and Hinden, R. (1997). *Internet Protocol, Version 6 (IPv6) Specification.* draft-ietf-ipngwg-ipng-spec-v2-01.txt. Internet Draft.

[Golombek et al., 1997] Golombek, M. P., Cook, R. A., Economou, T., Folkner, W. M., Haldemann, A. F., Kallemeyn, P. H., Knudsen, J. M., Manning, R. M., Moore, H. J., Parker, T. J., Rieder, R., Schofield, J. T., Smith, P. H., and Vaughan, R. M. (1997). Overview of the Mars Pathfinder Mission and Assessment of Landing Site Predictions. *Science*, 278(5344):1734–42.

[Iazzetta and Kon, 1995] Iazzetta, F. and Kon, F. (1995). MaxAnnealing: A Tool for Algorithmic Composition Based on Simulated Annealing. In *Proceedings of the Second Brazilian Symposium on Computer Music.*

[Kon et al., 1998] Kon, F., Campbell, R. H., Tan, S., Valdez, M., Chen, Z., and Wong, J. (1998). A Component-Based Architecture for Scalable Distributed Multimedia. In *Proceedings of the 14th International Conference on Advanced Science and Technology (ICAST'98)*, pages 121–135, Lucent Technologies, Naperville.

[Kriese and Tipei, 1992] Kriese, C. and Tipei, S. (1992). A Compositional Approach to Additive Synthesis on Supercomputers. In *Proceedings of the 1992 International Computer Music Conference*, pages 394–395, San Jose, CA.

[Leslie et al., 1996] Leslie, I., McAuley, D., Black, R., Roscoe, T., Barham, P., Evers, D., Fairbairns, R., and Hyden, E. (1996). The Design and Implementation of an Operating System to Support Distributed Multimedia Applications. *IEEE Journal on Selected Areas in Communication*, 14(7):1280–1297.

[Loy, 1985] Loy, C. (1985). Musicians Make a Standard: The MIDI Phenomenon. *Computer Music Journal*, 9(4):9–26.

[Moore, 1987] Moore, F. R. (1987). The Dysfunctions of MIDI. In *Proceedings of the 1987 International Computer Music Conference*, pages 256–263, San Francisco.

[MPEG.ORG, 1998] MPEG.ORG (1998). *MPEG-Audio layer 3 resources.* URL: `http://www.mpeg.org/index.html/mp3.html`.

[Puckette, 1994] Puckette, M. (1994). Is there life after midi? In *Proceedings of the 1994 International Computer Music Conference*, page 2, Aarhus, Denmark.

[Puckette et al., 1998] Puckette, M., Sorensen, V., and Steiger, R. (1998). The Global Visual Music Project. Project Home Page: `http://www.earunit.org/gvm.htm`.

[Sporer et al., 1992] Sporer, T., Brandenburg, K., and Edler, B. (1992). The Use of Multirate Filter Banks for Coding of High Quality Digital Audio. In *Proceedings of the 6th European Signal Processing Conference*, pages 211–214.

[Steinmetz and Nahrstedt, 1995] Steinmetz, R. and Nahrstedt, K. (1995). *Multimedia: Computing, Communications, and Applications.* Prentice Hall, New Jersey.

[Tan et al., 1996] Tan, S., Campbell, R., Chen, Z., Liao, W., Raila, D. K., Kon, F., and Valdez, M. (1996). Adaptation and Synchronization in Low-Bandwidth Internet Video. In *World Wide Web Consortium Workshop on Real Time Multimedia and the Web (RTMW '96)*, INRIA Sophia Antipolis, France.

[University Corporation for Advanced Internet Development, 1998] University Corporation for Advanced Internet Development (1998). Internet2 Project Home Page. `http://www.internet2.edu`.

[Vary and et al, 1988] Vary, P. and et al (1988). A speech codec for the European mobile radio system. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 227–230.

[Vercoe, 1997] Vercoe, B. (1997). *Csound Manual.* Media Lab, MIT. Edited by Jean Piché.

Acknowledgments

The authors gratefully acknowledge the support provided by CAPES and FAPESP. Fabio Kon is supported by a grant from CAPES, proc.# 1405/95-2 and Fernando Iazzetta is supported by FAPESP, proc.# 97/01688-1.