Wide Area Audio Synchronisation

Paul Barham and Richard Black and Ian Pratt
University of Cambridge Computer Laboratory
Cambridge CB2 3QG
United Kingdom

November 1995

1 Introduction

In this short paper we describe a Wide Area Audio Synchronisation demonstration using the Quality of Service (QoS) facilities of the Nemesis operating system. The Nemesis operating system is a result of the ESPRIT funded Pegasus I research project.

A stereo audio stream is split with each channel being input to one of two distinct network digitisation devices; one of the resulting streams is sent via a wide area network to a foreign country, and the two streams are then resynchronised entirely in software. Playout on a digital speaker device was analysed and found to be correct to within 150 μs, which is the resolution of the timer hardware on the workstation used. The accuracy of this resynchronisation demonstrates the effectiveness of the Nemesis QoS techniques.

2 Nemesis

General purpose multimedia computing platforms should endow text, images, audio and video with equal status: interpreting an audio or video stream should not be a privileged task of special functions provided by the operating system, but one of ordinary user programs. Support for such processing on a platform on which other user applications are running, some of which may also be processing continuous media, cannot be achieved using existing operating systems – it requires mechanisms that will consistently share out resources in a manner determined by both application requirements and user preferences.

Continuous media streams have two important properties. The first property is that their fidelity is often dependent upon the timeliness with which they are presented. This temporal property of continuous media imposes the requirement that code which manipulates the media data may need to be scheduled within suitable windows of time. The second property is that they are often tolerant of the loss of some of their information content, particularly if it is known how the data is to be used (e.g., many compression schemes rely on human factors to achieve high compression rates). This informational property, without regards to its exact nature, may be exploited by systems which handle continuous media.

The properties of the streams can be extended to the applications which process them; we have temporal requirements for such applications which are stronger than traditional data processing applications, and informational requirements which are weaker.

In order for an operating system to support both traditional and multimedia applications, a wider range of facilities than is found in current operating systems needs to be provided. The operating system which is being used for this Wide Area Audio Synchronisation work, and more generally, is the Nemesis operating system which is described more fully in [3, 2].

The main theme guiding the design of Nemesis is multiplexing system resources at the lowest level - in the case of the processor, this multiplexing system is the scheduling algorithm. However, it is the multiplexing of all resources, real or virtual, which has determined the fundamental structure of Nemesis.

This has given rise to a system in which as much functionality as possible executes in the domain of the application. This includes code that in a traditional microkernel would execute in a shared server. It should be emphasised that this need not change the interface seen by application programmers. The API seen by a programmer is often a thin layer of library code supplying a veneer over a set of kernel traps and messages to server processes - whereas in Nemesis the majority of the functionality would be provided by a shared library. As an example, a POSIX API in a Nemesis domain can be provided over a POSIX emulation which mostly runs within the application's domain.

In Nemesis a service is provided as far as possible by shared library code and the design of a service will aim to minimise the number of changes in protection domain. To aid in the construction of such services, references between the various parts of the code and data are simplified by the use of a single address space with protection between domains provided by the access control fields of address translations.

For more details about various aspects of Nemesis please see [3, 2, 1, 5].

3 Experimental Setup

The experimental equipment uses three workstations running the Nemesis operating system and three networked audio digitisation and playback devices. The audio CODECs used were the AVA200 [4] devices (now sold by Fore Systems Inc.) which attach, with the workstations, to a best-effort ATM network. The arrangement of the equipment can be seen in figure 1.

A conventional CD player was used to source an analog stereo signal, and the left and right channels were fed into different AVA200 devices and digitised at 44.1kHz. The final stereo stream generated was played out on an ATV300 (from the same manufacturer) to stereo headphones, which can be listened to by the observer, and also to a two channel digital oscilloscope which can be used to determine the exact timing difference between the left and right audio channels.

The problems which this experiment imposes on our software audio synchronisation technique, and our QoS operating system are several.

- The two different audio capture devices are operating on their own crystal oscillators, each of which may be up to 50ppm from the nominal frequency. Therefore samples may be being generated at different rates.
- One of the channels is taking a network path with an additional mean latency of 20ms and very substantial jitter characteristics. In addition, this channel suffers from very occasional data loss.
- The final playout device is again operating on its own free running crystal oscillators, which may be up to 50ppm from its nominal frequency. Therefore the playout device may be consuming samples at a rate which may be slower than both the sources, slower than one of the sources and faster than the other, or faster than both of the sources.

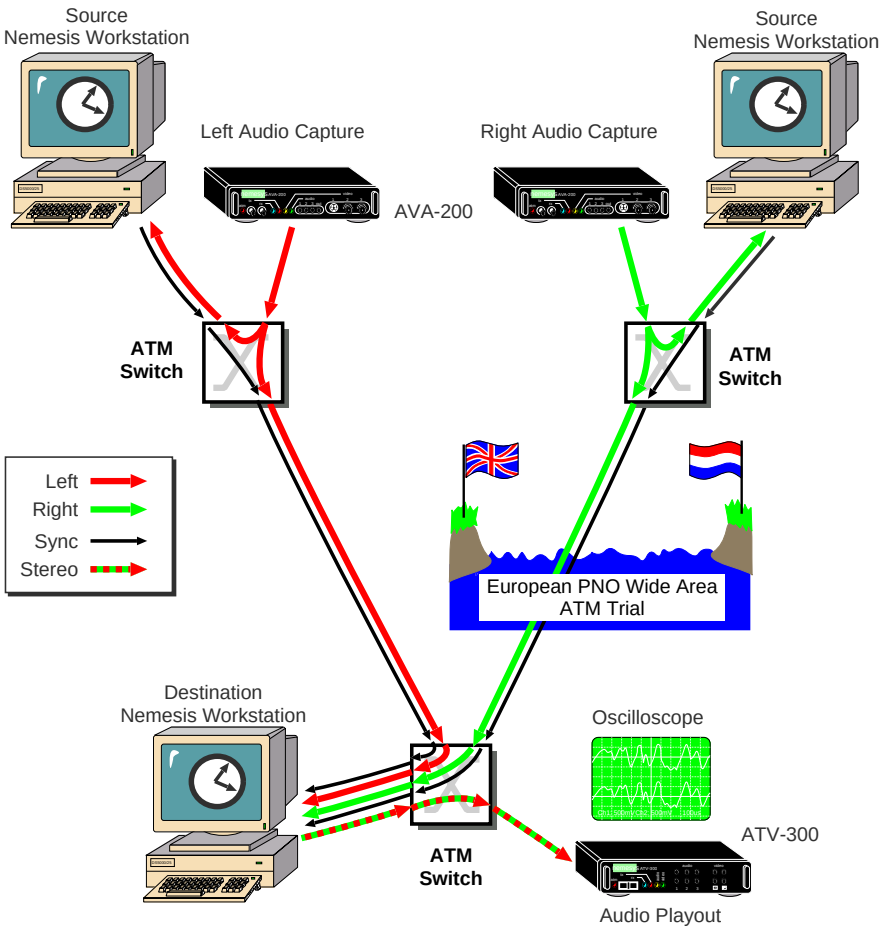

Figure 1: Audio Stream Routes

- The streams must be merged on a standard workstation whose clock resolution is 122.5 μs

Fortunately, the individual Audio codec devices had been designed to the best practice available in the published literature (particularly [6]). For example, the AVA200 devices place a sample sequence number in the audio packets that they emit. This made the overall system problem tractable in software, which we will now describe.

4 Audio Software Techniques

Conceptually there are several component to solving this problem in software. Each of which will now be described.

4.1 Generating the Synchronisation streams

The key to the problem is to know which left ear sample corresponds with which right ear sample. There will be “slippage” between these two sample streams due to the differing sampling clocks so this correspondence will change during the lifetime of the experiment. Note that it is only necessary

to have the samples referenced to a "time" which is common amongst the set of sources which must be synchronised. This "reference time" is only logical, it need neither be global nor perfect.

In our experiment a Nemesis workstation is placed on the same network as each audio source. The ATM circuit is organised so that a copy of the audio is provided to the Nemesis workstation adjacent to the source as well as to the network towards the destination. This can be seen at the top of figure 1. Since this is in the Local Area no loss occurs between the audio capture and the workstation adjacent to capture, and it can be reasonably assumed that the delay is both small and fixed.

The AVA200 devices place a sample sequence number in the packets of audio samples they emit. When the packets reach the workstation, the Nemesis operating system immediately timestamps the packets in the kernel with that local system time at which they arrived.

When the software application running on the workstation receives the packet, it therefore knows the local time at which a particular sequence number in the audio stream was generated. The software performs a calculation to determine the reference time which corresponds to the local time, and hence the reference time which corresponds to the sample number in the stream.

At periodic intervals, the software in the source Nemesis workstation emits packets on a synchronisation connection (labeled "Sync" in figure 1) which lists the association between reference time instants and sample sequence numbers. These sync packets travel across the network to the destination machine. Note that the sync packets may suffer a different network delay, jitter, and loss to the audio packets which they refer to.

4.2 Providing Reference Time

Reference time between the two audio sources may be provided by any means which is convenient. For example, NTP may be used to provide reference time from a remote accurate clock across the network. Alternatively NTP-like protocols could be used entirely intra-domain between the two source workstations to cause reference time to progress at the mean of their workstations' own local clocks.

Note that full NTP would be required here, and Simple NTP as found in certain software such as Microsoft Windows would be insufficient.

4.3 Merging the streams

We also place a Nemesis workstation adjacent to the destination audio hardware device. This workstation receives the four streams from the network (two audio and two synchronisation), and generates the stereo output stream for the ATV300. This is shown at the bottom of figure 1.

The ATV300 contains a small playout buffer to cope with samples being delivered in bursts (packets), packetisation delays, and jitter in the local area. It monitors the long term average occupancy of its own playout buffer against its own oscillator which is clocking the consumption of the samples. It can therefore determine whether the samples arriving in the packets are being delivered at a rate which is faster or slower than its own local oscillator. It takes action to match these rates by occasional sample deletion or interpolation as appropriate. This action handles any clock differences between the ATV300 and the destination Nemesis workstation.

The destination Nemesis workstation maintains its own playout buffer to account for the latency and jitter in the wide area network connection. By measuring the mean and variance of inter-arrival time of packets from the network it calculates the required target occupancy (length) of this buffer to provide a very high probability that it will be able to deal with network effects. In the environment of figure 1 this means that the left ear samples, which are delivered in the local area, are buffered for many milliseconds.

The workstation monitors the long term average progression of source reference time (arriving in the source synchronisation streams) against its own local time, to determine the correspondence. It can then use its own local scheduling time to determine when to send packets of samples to the ATV300 playout device.

To create the stereo samples the destination knows what reference time it would like to create a sample for. It can use the two synchronisation streams to map from reference time to sample number, and the sequence numbers in the audio streams to select the audio samples. These can then be combined and sent to the playout device.

In addition, since all the packets are sequenced, the destination Nemesis workstation can adapt to packet loss in the network. Loss of synchronisation packets can be handled by assuming historical linear progression until the next synchronisation packet arrives. Loss of audio data can be handled by gross interpolation, by interpolating to silence, silence insertion, and interpolating to subsequent received data, or by copying the samples from the other ear's channel.

5 Results and Conclusions

Using an oscilloscope the total latency of the audio streams was measured at under 30ms and the streams were observed to be synchronised to an accuracy of better than 150 μs (the system clock resolution on the workstations used was 122.5 μs). Even if the computers' clocks had not been the restriction, 150 μs represents only seven samples, or two inches in free air.

In this experiment there are seven clocking domains. One in each piece of capture hardware, one in each Nemesis workstation at source, the source reference clock, the clock in the Nemesis workstation at the destination, and the clock in the playout hardware device. The overall problem is solved through our software audio synchronisation techniques, and the high QoS for scheduling and network access guarantees obtainable throught the use of our Nemesis operating system.

References

[1] Richard J. Black. Explicit Network Scheduling. Technical Report 361, University of Cambridge Computer Laboratory, December 1994. Ph.D. Dissertation.

[2] I. M. Leslie, D. McAuley, R. Black, T. Roscoe, P. Barham, D. Evers, R. Fairbairns, and E. Hyden. The design and implementation of an operating system to support distributed multimedia applications. *IEEE Journal on Selected Areas In Communications*, 14(7):1280–1297, September 1996. Article describes state in May 1995.

[3] Sape Mullender, Ian Leslie, and Derek McAuley. Operating System Support for Distributed Multimedia. In *USENIX*, pages 209–219, June 1994.

[4] Ian Pratt and Paul Barham. The ATM Camera V2: AVA200. In Richard Black, editor, *ATM Document Collection 3 (The* Blue *Book)*, chapter 36. University of Cambridge Computer Laboratory, March 1994. Available from `http://www.cl.cam.ac.uk/Research/SRG`.

[5] Timothy Roscoe. The Structure of a Multi-Service Operating System. Technical Report 376, University of Cambridge Computer Laboratory, August 1995. Ph.D. Dissertation.

[6] Cormac J. Sreenan. Synchronisation services for digital continuous media. Technical Report 292, University of Cambridge Computer Laboratory, March 1993. Ph.D. Dissertation.