Network Music Performance

Zefir Kurtisi

Institut für Betriebssysteme und Rechnerverbund
Technische Universität Braunschweig

March 13, 2007

Outline

About

Goals

Enable distributed Network-centric Music Performances for E-Learning and recreational purposes using

- today's hardware
- today's networks

Initialization

- Idea by Xiaoyuan Gu
- Initialized as a practicum for students in the summer term 2004

Progress

- practicum version proved the concept
- recent work led to stable prototype working in fast WANs (like DFN)
- future work on movement towards the home user

Overview

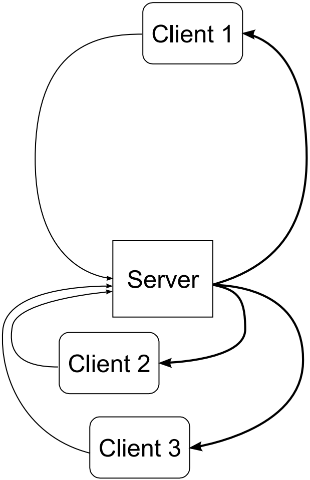
Features

Meeting above Goals requires support for

- Realtime Rehearsal
- Rehearsal On-Demand

Basic Principle

Centralized NMP-Server synchronizes and mixes audio data of all participating Clients and returns mix with minimal latency.



Classification

NMP is an interdisciplinary field between

- Music
- Psychology
- Physics
- Engineering
- Computer Science

Contribution

We focus our work in some areas of Computer Science like

- Networks
- Hardware evaluation
  - Audio and Network cards, Audio Equipment
- Audio Processing
  - Audio Manipulation, Compression, Mixing and Synchronization
- Realtime critical aspects of Operating Systems
  - Content Switching, Thread Scheduling
  - Audio APIs and Frameworks

Critical Requirement

Musicians' Demand

Music and Psychology shows that musicians' delay tolerance is very low and depends on

- Musician (skill, personality) and Instrument
- Composition and Music style

As a rough estimation a total delay of 30ms should be provided.

Technical Challenge

To be practically relevant NMP should not exceed this delay in total, including

- delay in the end-systems (client and server) for
  - processing and buffering
- network latency including
  - propagation time and de-jittering

Related Work

Is this not the same like...

- VoIP? $\Rightarrow$ P2P, no mixing and syncronization, higher delay-tolerance
- Internet Audio Conferences? $\Rightarrow$ same as VoIP
- Network Games? $\Rightarrow$ no packet-based mixing and synchronization

Similar Work

- Internet2
  - HYDRA and the Miro Quartet (imsc-USC)
  - SoundWire (CCRMA-Stanford)
  - Transatlantic Master Class
- MIDI-based
  - *VirJa* Virtual Jazz Session System
  - eJaming
- NMP (Alexander Carot) $\Rightarrow$ P2P, no mixing

Objectives

Main goal

Provide a system that operates with a maximum total delay of 30ms.

Results

Some results and facts not discussed here

- best results using 48kHz sampling rate at 16bps with 128 sample-blocks
- there are no realtime capabilities in desktop operating systems
  - Windows shows the worst reactivity, threads stall for 100s of ms
  - Mac OS/X is significantly better, threads stall up to 50ms
  - Linux is best, threads stall for up to 30ms, improvable through RtLinux or RTAI
- RtAudio is used as platform independent Audio API
- packet losses degrade audio quality
  - FEC and error concealment strategies are mandatory
- frequency transforming audio codecs introduce too large buffer delays, we are restricted to
  - uncompressed audio (PCM)
  - nonlinear sample-based requantization (ADPCM)
  - lossless non-transforming entropy coding (FLAC)

Delay Analysis in the Endsystems

Delay Sources

- hardware buffers
- buffers for data transformation
- processing time (CPU cycles)
- resource sharing (scheduling)
- system stalls (periphery IO)

Buffer Delay Granularity

Buffering delay granularity is set by audio card buffering through

- buffer size, here 128 samples/buffer
- sampling frequency, here 48kHz

$\Rightarrow$ results in buffering delay $t\varphi = 2.66ms$

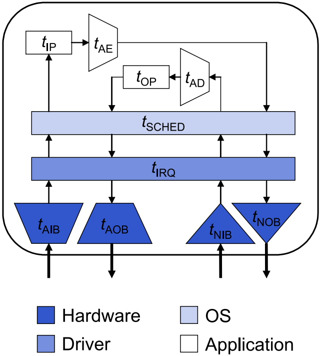
Client Delay

The client has to pass and process data from one hardware device to another.

Minimal latency

Assuming negligible processing delay, the minimum buffering delay is

- one audio block in the audio card's playout buffer (DAC)
- one audio block in the audio card's record buffer (ADC)
- one audio block in the audio card's driver for output
- one audio block for processing and network de-jitter



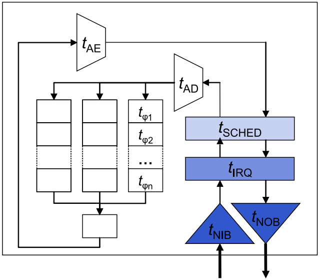
Result

The client's lower delay bound is $4 \cdot t\varphi$, here 10.66ms

Server Delay

The server has to

- de-jitter incoming audio packets
- synchronize all audio streams
- mix audio data (linear or positional)
- send mixed audio to each client



Minimal latency

Assuming ideal network conditions with no jitter

- no de-jitter buffer is needed
- one audio buffer is needed for synchronization

$\Rightarrow$ The server's lower delay bound is $t\varphi$, here 2.66ms

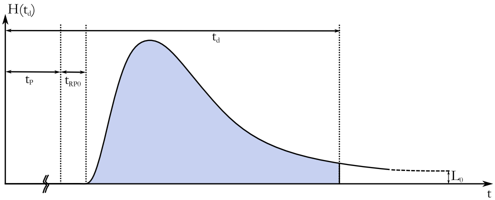
Conclusion

Client and Server require a minimum delay of $5 \cdot t\varphi$, here 13.33ms

Network Delay

Restrictions

- no QoS
- no retransmissions
- late packets are lost packets



Result

- we can estimate propagation delay upon network distance
- we can not estimate or predict total network delay

$\Rightarrow$ no direct control over network, we can only react

Musicians' Tradeoff

Tolerable Audio Quality Degradation $\Leftrightarrow$ Tolerable Latency

Conclusion

NMP should be possible in a stable network and could potentially connect musicians up to 500km apart (round-trip).

Conclusion

NMP should be possible in a stable network and could potentially connect musicians up to 500km apart (round-trip).

Reality Check

- works in the institute's network with latencies down to 13.3ms
- works in the DFN (German Research Network) between IBR and Lübeck (500km round-trip)

Some Measurements

Type	Network Buffer		Total Delay	Packet Loss Statistics		
	[packets]	[ms]	[ms]	mean	max	dev
LAN	0	0	13.3	0.060%	2.4%	0.173%
LAN	2	5.3	18.7	0.015%	1.6%	0.101%
WAN	6	16.0	29.3	0.689%	6.9%	1.254%
WAN	10	26.7	40.0	0.216%	3.8%	0.519%

Summary

Status so far

- working prototype has been developed
- theoretical delay bounds have been achieved
- realtime performances are practicable in fast broadband networks

Open Issues

- many interdisciplinary issues
  - do musicians prefer constant or adaptive latencies?
  - define parameter sets for different combinations of musician, instrument, style, etc.
- audio compression
  - can we achieve better data reduction if we knew instrument, etc.
- can we expand NMP to larger scale networks (e.g. to home users)
  - which impact has the lower (uplink) network capacity of e.g. ADSL
  - are ISP networks stable enough for this application
  - how to cope with the triangular routing problem
- ...

Thank You!

Questions?

<kurtisi@ibr.cs.tu-bs.de>