A Low-Latency Full-Duplex Audio over IP Streamer

Asbjørn Sæbø and U. Peter Svensson
Centre for Quantifiable Quality of Service in Communication Systems (Q2S)
LAC2006, 2006-04-27
What is LDAS?

— A Low Delay Audio Streamer in software
  • basically an audio to UDP/IP adapter
  • developed for Linux, using the ALSA sound system
  • currently at prototype stage
— A research tool
  • distributed multimedia interaction
  • perceived quality of service
— Aimed at demanding applications
  • low latency, high quality multichannel audio
Overview of the presentation

— Background and motivation
— Requirements and specification
— Main points of LDAS implementation
— Latency and latency measurements
— Conclusion
Q2S: Audio over IP Networks

— Quality beyond voice over IP:
  • Lower latency
  • Higher audio quality $\Rightarrow$ higher bit rates
  • Multiple channels $\Rightarrow$ higher bit rates

— Existing solutions
  • No access to source code, and/or
  • Not fulfilling requirements
  • Much to learn from “rolling our own”

— Goal: Fully open software, suitable as research tool
Worldview

Quality of service, as perceived from the endpoints, by the users. Connected by the net: Virtual presence and true interaction
Networked ensemble playing

Other applications:
— Transmission of acoustic environments
— Advanced video conferencing
Related work at Q2S/NTNU

Ola Strand:
  — Transmission of audio and video

Håkon Liestøl Winge:
  — Measured influence of latency

Otto Wittner / Sigurd Saue:
  — Low-delay Windows Streamer

Trond Iver Røste Pedersen:
  — Streaming in Java

Snorre Balliere Farner et.al.:
  — Influence of latency and reverberation
Requirements

— Low latency:
  Less than 20 - 30 ms for ensemble playing

— Multiple channels:
  For realistic transmission of
  acoustic environments and multi-channel 3D audio

— High quality:
  CD-quality and upwards
Latency: How low can it be?

- Trondheim - New Zealand: 170 ms

- The speed of light – too low?
- Only “near” parts of the world within reach
Specification (minimum)

**Audio:** 44.1kHz/48kHz, 16bit, two-channel PCM, possibility for coding.

**Network:** UDP/IP, with precautions against UDP shortcomings. No retransmission.

**Latency:** Less than 20 ms (analog to analog) over LAN.

**Synchronisation:**
- Receiver and sender must be kept in sync.
- Buffering, to minimise the effects of network transmission time jitter.
Implementation overview

Three threads: Recorder/sender, receiver and playback

- Read audio data from sound card
- Build packet, send to network
- Process audio
- Receive packet from network
- Enqueue packet from enqueued data
- Build audio period
- Write audio data to sound card
- Process audio
Packet format

The payload of the UDP packet:

<table>
<thead>
<tr>
<th>Audio data (one period)</th>
<th>Seq. Num.</th>
<th>Time stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame 1</td>
<td>Frame 2</td>
<td>Frame M</td>
</tr>
<tr>
<td>Ch. 1 sample</td>
<td>Ch. 2 sample</td>
<td>Ch. N sample</td>
</tr>
</tbody>
</table>

Packet stream positions:
Sequence number plus frame-level offset into packet
Packet stream control

Protocol implicitly defined by packet format (sequence numbering) and receiver enqueueing combination. Handles UDP shortcomings and network transmission problems.

- Order packets
- Detect lost/missing packets
- Reject duplicates
- Reject late packets

Missing packets replaced by dummy data. (Possibility: Error concealment.)
Synchronisation

— The receiver queue
  • A sliding window onto the packet stream
  • Ringbuffer of pointers to packets

— “Large scale” synchronisation
  • “Early” packet (outside window) ⇒ resynchronisation
  • “Too many” late packets recently ⇒ resynchronisation

— Drift adjustment by watermark algorithm
  • Queue length too high ⇒ skip single samples (frames)
  • Queue length too low ⇒ reuse single samples
Latency measurements

Audio delivered in *periods*
- Sender sound card buffer: One period of latency
- Receiver sound card buffer: Up to one period
- Receiver queue: Adjustable
- Network transmission time
- A/D and D/A: 2 – 3 ms
- Processing latency (software, OS)
Measurement setup

- Two Linux computers, 2.6.12 multimedia kernels
- M-Audio Delta44 audio interfaces
- Connected through a switch
- Full duplex stereo transmission with LDAS
- Impulse response measurements
- Transmission monitored by listening to audio
Impulse response measurements

![Impulse response graph](image-url)
Frequency response

![Frequency response graph]

- Frequency [Hz]: 10, 100, 1000, 10000
- Pressure [Pa]/[volts], [dB]: 70, 65, 60, 55, 50, 45, 40, 35, 30, 25, 20

WinMLS Pro

Sæbø and Svensson, Low-Latency Audio Streamer
## Latency results

<table>
<thead>
<tr>
<th>P. size</th>
<th>Queue len.</th>
<th>Mean latency</th>
<th>St. dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>3</td>
<td>15.2</td>
<td>0.8</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>11.0</td>
<td>0.8</td>
</tr>
<tr>
<td>128</td>
<td>0.1</td>
<td>8.2</td>
<td>0.6</td>
</tr>
<tr>
<td>64</td>
<td>1</td>
<td>5.8</td>
<td>0.4</td>
</tr>
<tr>
<td>64</td>
<td>0.1</td>
<td>4.9</td>
<td>0.4</td>
</tr>
<tr>
<td>32</td>
<td>0.1</td>
<td>3.1</td>
<td>0.2</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>2.6</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Latency in milliseconds for audio transmission with LDAS. (Measured from analog input to analog output. Fs = 48kHz.)
Similar solutions

— netjack (Torben Hohn)
— jack.udp (Rohan Drape)
— streamBD (SoundWIRE group, CCRMA)
— llcon (Volker Fischer)
Thanks!

In particular:
- Lee Revell, Mindpipe Audio

Advice and discussions:
- Svein Sørsdal, SINTEF
- Georg Ottesen, SINTEF
- Jon Kåre Hellan, Uninett
- Paul Calamia, Rensselaer Polytechnic Institute
Availability

— Open Source, GNU General Public License
— Download: http://www.q2s.ntnu.no/~asbjs/ldas/ldas.html
— Mailing list: https://pat.q2s.ntnu.no/mailman/listinfo/ldas-dev