REAL-TIME HIGH-QUALITY AUDIO STREAMING

Juan-Pablo Cáceres
Network Musical Performance Workshop
Technical and Artistic Strategies to Perform Around the Globe
Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
Perform Music On-Line in Real-Time with the Highest Audio Quality Possible
Goals of High-Quality Audio over Networks

**Maximize Audio Quality** for available networks conditions

**Minimize Latency**

But more important, **Minimize Jitter**

Adjustable **Number of Channels**

**Audio routing flexibility**

**Multiple peers**
Keep Delay **Constant**
Maximize Audio Quality
Why Latency Matters

Saint Lawrence String Quartet (Quintet)

Banff Centre, Alberta, Canada
Quartet

25 ms One-way Delay

Stanford, Anechoic Room
Viola
Why Latency Matters

Saint Lawrence String Quartet (Quintet)

25 ms One-way Delay

‣ What Happens Naturally with Delay?
‣ Now, they are conscious...
‣ ...and they try to be stable
‣ The whole Quintet
Some Historical/Technical Foundations

Basic Principle: Uncompressed Audio

Year 2000

McGill  Xu & Cooperstock
Stanford  Chafe, Wilson, Leistikow, Chisholm, Scavone
## UDP and TCP revisited

<table>
<thead>
<tr>
<th>UDP</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Unreliable</td>
<td>- Reliable</td>
</tr>
<tr>
<td>- Connections-less</td>
<td>- Connected (virtual circuit)</td>
</tr>
<tr>
<td>- Datagram Oriented</td>
<td>- Byte-Stream Oriented</td>
</tr>
</tbody>
</table>

**Delay Constant?**
Why TCP is problematic

Transmission Control Protocol (TCP) is a widely used protocol for communication over the Internet. It provides a reliable, end-to-end service for application processes, ensuring that data is delivered accurately and in order. However, this reliability comes with some limitations and complexities.

TCP manages data flow using byte streams. A byte stream is a sequence of bytes that flows in both directions between two endpoints. TCP uses a window mechanism to control the flow of data. The window is a sliding window that separates the sequence of bytes into smaller units.

When a process wants to send data, it writes bytes into its send buffer. The process is responsible for writing the bytes into the buffer. TCP then sends segments containing the bytes over the network. Each segment contains a header with fields that uniquely identify the connection, such as the source and destination ports.

The receiver reads bytes from the TCP connection and stores them in a receive buffer. The receiver must ensure that it reads bytes in order and acknowledges the receipt of each segment. If a segment is lost or delayed, the receiver will not acknowledge it, and TCP will retransmit the segment.

TCP also uses a three-way handshake to establish a connection. This handshake involves the exchange of initial packets to confirm that both ends are ready to communicate. Once the connection is established, data can be transmitted over the connection.

TCP's reliability comes at a cost. It requires additional processing to ensure that data is delivered accurately. This adds latency to the network, which can impact performance, especially for real-time applications. Additionally, the window mechanism can lead to congestion if not managed properly.

In summary, while TCP provides a reliable service for communication, it also introduces overhead and complexity. Understanding these aspects is crucial for optimizing network performance and application design.
TCP makes the underlying delay elastic and ever-increasing

Use UDP
A JackTrip Session
What is Jack?

Demo: Jack
Real-Time Audio: Under the Hood

Audio

Out

In

UDP3  UDP2  UDP1

UDP1  UDP2  UDP3

Audio

In

Out
Real-Time Audio: Under the Hood

Audio Inputs (N-Channels) ➔ Audio Process Callback
  JackAudio Threads ➔ Process plug-ins ➔ Non-blocking

Audio Outputs (N-Channels) ➔ Audio Process Callback
  Non-blocking ➔ Ring Buffer ➔ Blocking ➔ UDP Sender Thread ➔ To Peer
  Network

Non-blocking ➔ Ring Buffer ➔ Non-blocking ➔ UDP Receiver Thread ➔ From Peer
  Network
Real-Time Audio: Under the Hood

Ring Buffer initial "symmetric" pointer positions

Under-run Condition
- Silent Mode: Read packets of zeros
- Wavetable Mode: Loop through the last read packet

Overflow Condition
- Reset pointers and clear older packets
Real-Time Audio: Under the Hood

Sender end

```
UDP1
  p3  p2  p1
To Peer
```

```
UDP2
  p4  p3  p2
To Peer
```

```
UDP3
  p5  p4  p3
To Peer
```

Receiving end

```
UDP1
  p3  p2  p1
From Peer
```

```
UDP3
  p5  p4  p3
From Peer
```

Lost Packet
Parameters that matter

Queue Length (latency/jitter tradeoff)

Redundancy (bandwidth)

Audio Bit Resolution (bandwidth)

8 / 16 / 24 / 32
Parameters that matter

Packet size (latency)
Sampling Rate (bandwidth)
The smaller the packet size, the lower the latency (sampling rate constant)
Sampling Rate / Latency

For the same packet size

- **48kHz**: \(\frac{64}{48000} = 1.3\) ms
- **96kHz**: \(\frac{64}{96000} = 0.7\) ms

The higher the Sampling Rate, the lower the delay (packet size constant)
A Simple (default) JackTrip Session

at CCRMA (Server)

```
jacktrip -s
```

server mode

at UK (Client)

```
jacktrip -c [ccrma-IP-number]
```

client mode
More Control over a JackTrip Session

at CCRMA (Server)

\texttt{jacktrip -s -n 8 -q 4 -r 2}

\begin{itemize}
  \item server mode
  \item num chans
  \item Queue
  \item redundancy
\end{itemize}

at UK (Client)

\texttt{jacktrip -c \{ccrma-IP-number\} -n 8 -q 4 -r 2}

\begin{itemize}
  \item client mode
\end{itemize}
QoS Network “Audible Distances”

Plucking the Net

Longer Distance = Lower Pitch

Short Distance = Higher Pitch

Short Jittery/Lossy Wireless

JackTrip Server

client

client

client
Net vs. Net Collective

Chris Chafe | Chopper

Juan-Pablo Cáceres | CCRMA | Stanford University | jcaceres@ccrma.stanford.edu
What’s next for JackTrip

Windows XP, Vista Port | Elie Noune

Garry Scavone’s RtAudio
JamLink

http://www.musicianlink.com/
More Information

http://ccrma.stanford.edu/groups/soundwire/

JackTrip at Google Code:
http://code.google.com/p/jacktrip/

JackTrip Mailing List:
http://groups.google.com/group/jacktrip-users