

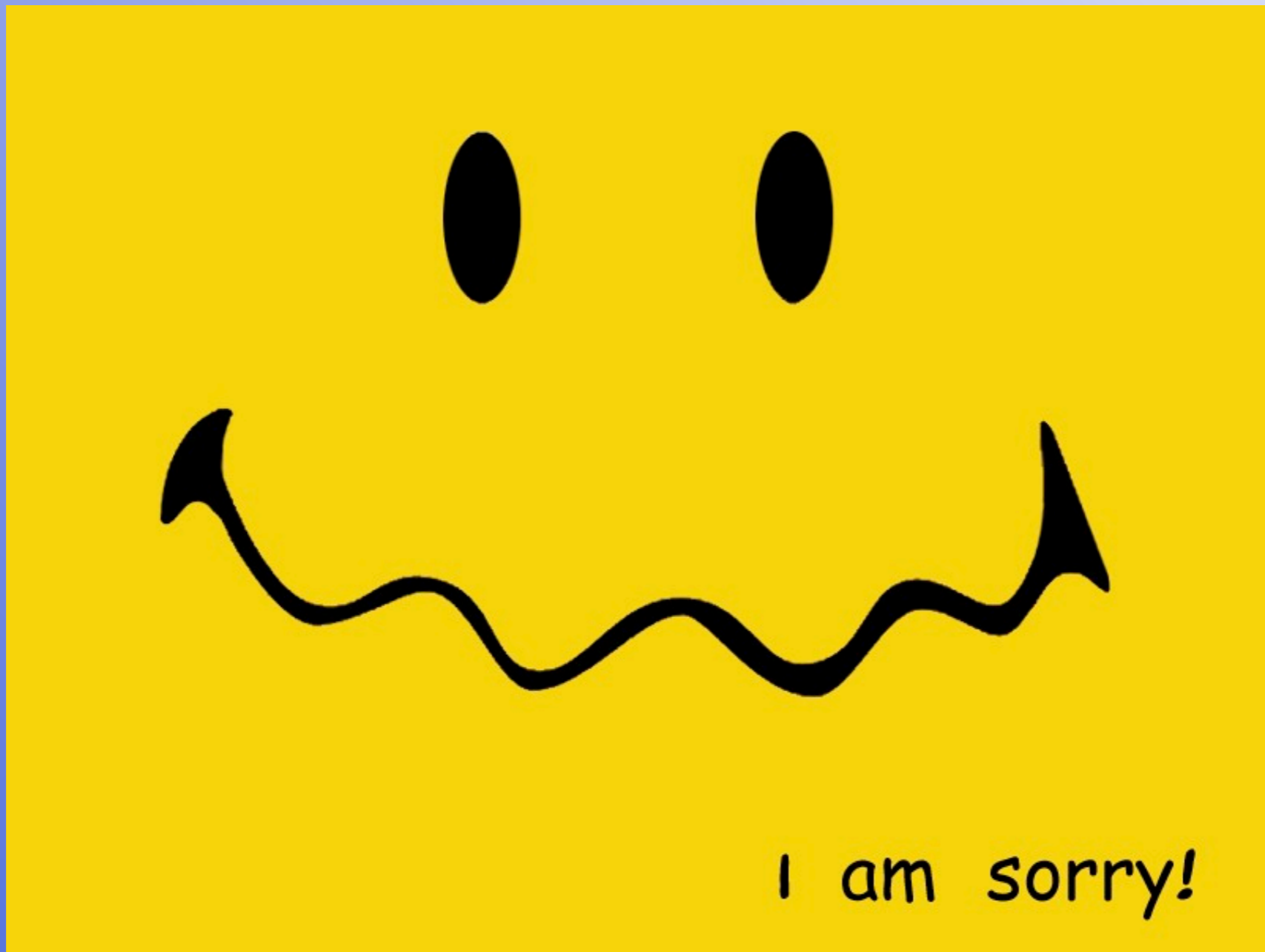


Towards a More Effective OSC Time Tag Scheme

Adrian Freed
adrian@cnmat.berkeley.edu
Research Director

UC Berkeley Center for New Music and Audio Technologies (CNMAT)
1750 Arch St.
Berkeley, CA 94709, USA
<http://www.cnmat.berkeley.edu>

Whoops



mature OSC features

- We had years of experience with all the other OSC features you know and love:
 - Namespaces
 - Regular expressions
 - Datagrams
 - Network Byte Order

What went wrong?

- The only OS (IRIX) affording a good implementation was becoming irrelevant
- Other scheduling issues dominated latency problems, i.e., Audio and MIDI I/O
- There was no standard cross platform NTP server
- Nobody bothered so implementations are incomplete so nobody bothered....

Time for Time-tags

- They are good for:
 - Synchronization of activity on different OSC server nodes
 - Synchronization when transport requires large updates to be fragmented
 - Jitter Attenuation
 - Simplify “Sequencer” applications
 - Record when things happened

Current Specification

- Time tags are 64-bit time used by NTP
- Servers schedule message processing to make the message update occur at the times in the tag
- Assumes the client and server clocks are synchronized

Issues

- It is still hard to find a cross-platform NTP server for ad-hoc LANs built for a particular performance
- Clients don't know how far forward to set the tags to attenuate jitter because they don't know what the network latency will be

Solutions 1/2

- Relativistic clocks: clients use a local 64-bit time in their tags; they include an additional tag that specifies when they sent the packet
- Servers compute time of arrival statistics to develop a slowly varying jitter advance to rewrite timetags:
 - $\text{new_time} = \text{original_time} - \text{sent_time} + \text{jitter_advance} + \text{arrival_time}$

Solutions 2/2

- If bidirectional channels are available an NTP-like scheme can be used so clients can synchronize to server clock. Clients tag their messages according to the clock approximated for each server
- It supports senders and receivers joining and leaving network dynamically and avoids need to identify an NTP master

Let's fix this formally

- Please contact me if you are interested in
 - joining the “OSC time-tags” working group
 - suggesting further issues and solutions
 - describing your needs
 - sponsoring this work
- adrian@cnmat.berkeley.edu