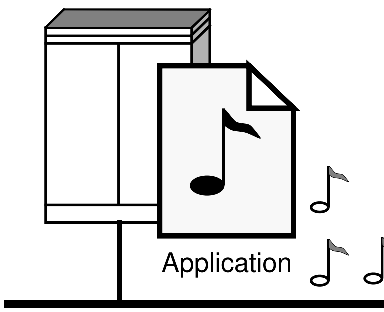
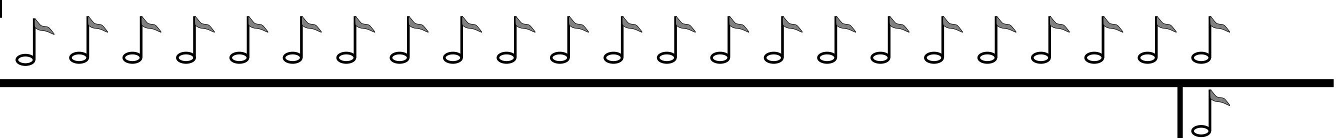
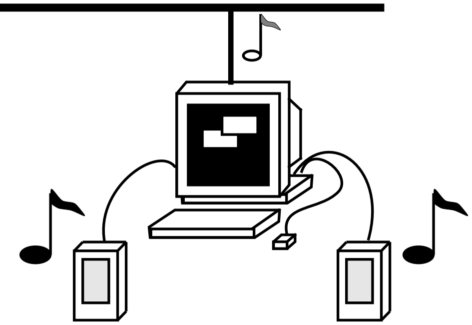
NetAudio

An Audio Protocol for Networks







Jim Fulton
Greg Renda
Dave Lemke
Network Computing Devices, Inc.
netaudio-spec@ncd.com

Goals of NetAudio

- Provide network access to audio I/O hardware
  - Network-transparent like the X Window System
  - Handle multiple applications at once, even accessing same devices
- Ease of Use – automatic conversions in server
  - Data formats – audio data and byte orders
  - Sample rates – resamples if multiple apps have different rates
- Allow applications to “wire up” inputs and outputs
  - Variety of sources and sinks: network, device, stored bucket, waveforms
  - Data can come from or go to any combination or sources/sinks
  - Splitting and merging of multiple tracks
  - Manipulate gain of any track

Primitives in NetAudio

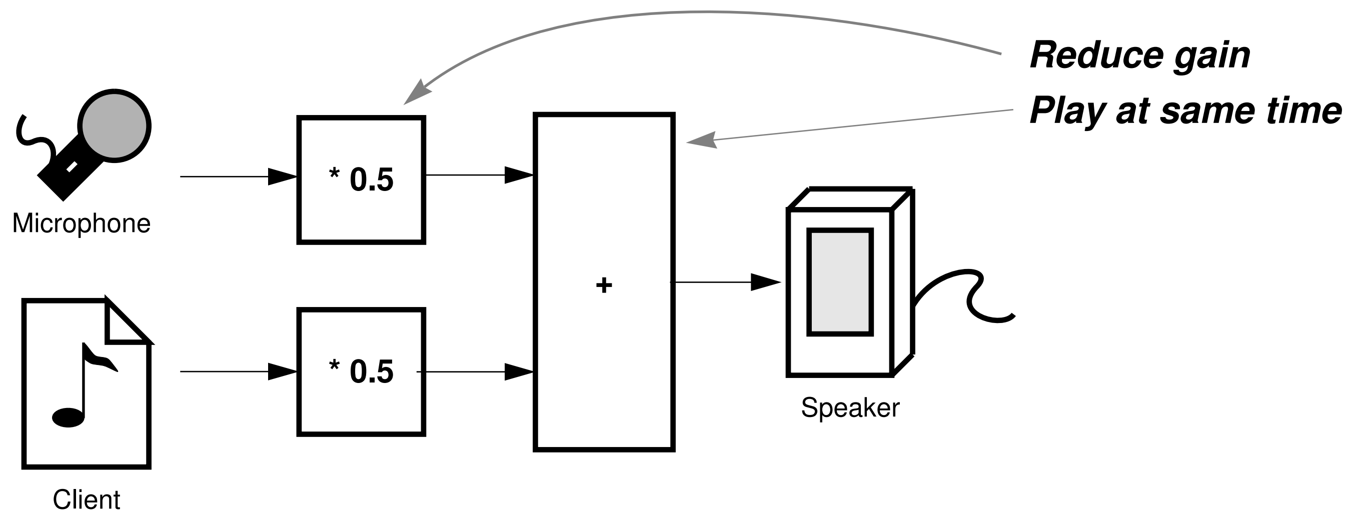
- Audio Data
  - Streams of numeric sample values [-1.0, +1.0]
  - Extensible set of encodings for values
  - Sample rate indicating granularity of stream
- Inputs and Outputs
  - Produce or consume a stream of audio data
  - Extensible set of virtual and physical devices
  - Devices, network, buckets (stored sounds), waveform generators
- Audio Flows
  - Pipeline of data from inputs, thru extensible set of operators, to outputs
  - Multiple tracks
  - Values manipulated through adders, multipliers, mixers, splitters
  - Client can set low- and high-water marks on inputs and outputs

Automatic Conversion

- Data Encoding Formats
  - Linear: signed/unsigned, 8/16 bit, LSB/MSB
  - µLaw
  - Automatically converted in flows and to destination format
- Sample Rates
  - Application may request data at any rate from 1 kHz to 50 kHz
  - Server asks devices for data at highest requested rate
  - Server resamples data for client
- Output Levels
  - Data from multiple flows mixed to avoid clipping

Flows

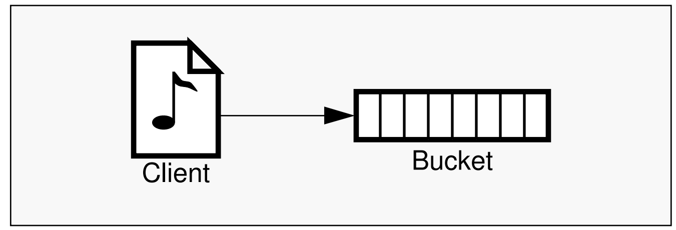
- Applications wire inputs, manipulators, and outputs



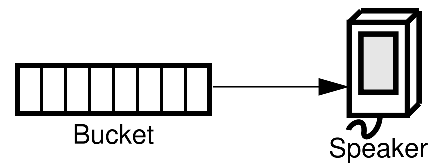
- Flows can be ***started***, ***stopped***, or ***paused***
  - Explicitly or triggered by presence/absence of data

Examples of Flows

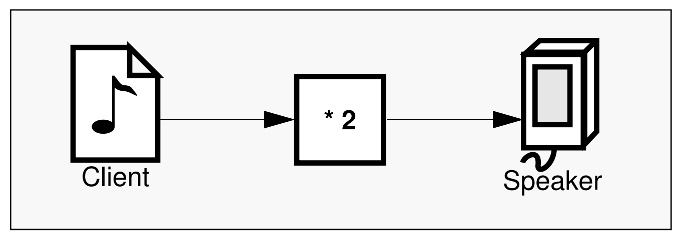
- Storing in Server
  - App can replay many times
  - Client specifies amount of data



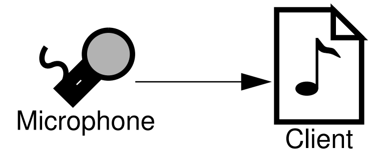
- Replaying stored sound
  - Very fast to start playing
  - Applications can **share** buckets



- Playing directly to speaker
  - Can send infinite amount of data
  - Client sends data in chunks

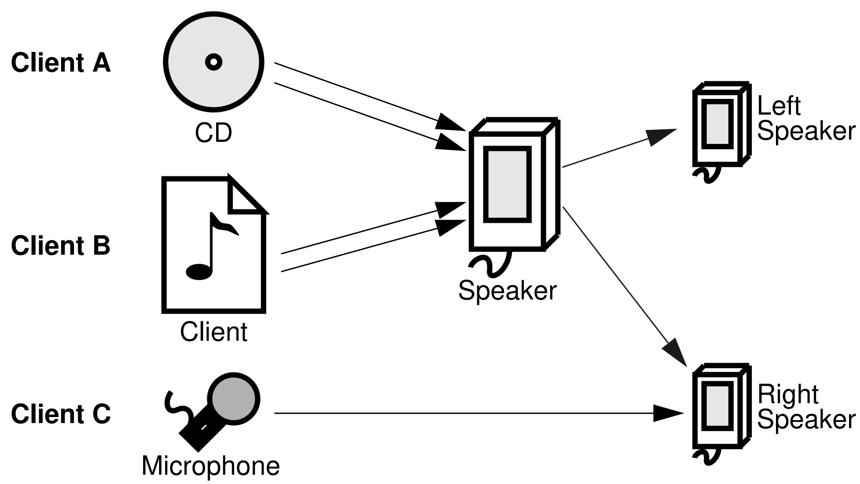


- Recording directly to Client
  - Server temporarily buffers data
  - Can fetch infinite amount of data



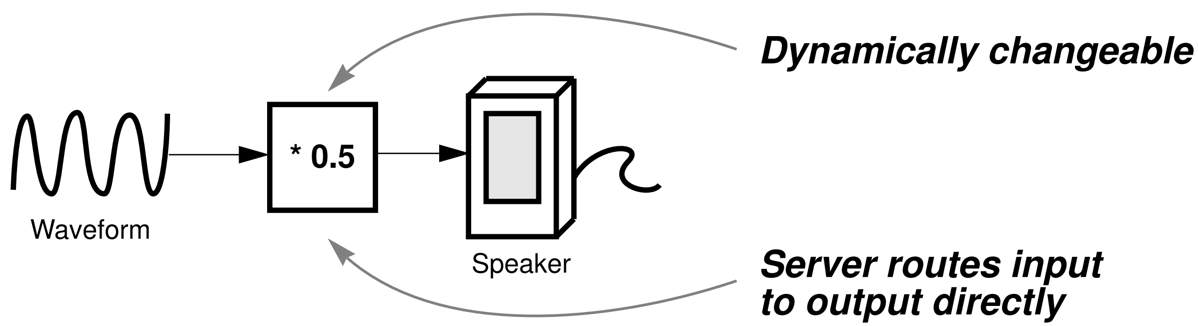
Multiple Tracks

- Data within a Flow can have up to 32 tracks
- Server can provide virtual devices built from others
  - Stereo speaker could use two individual mono speakers
  - Subchannels kept synchronized (if one pauses, so does the other)



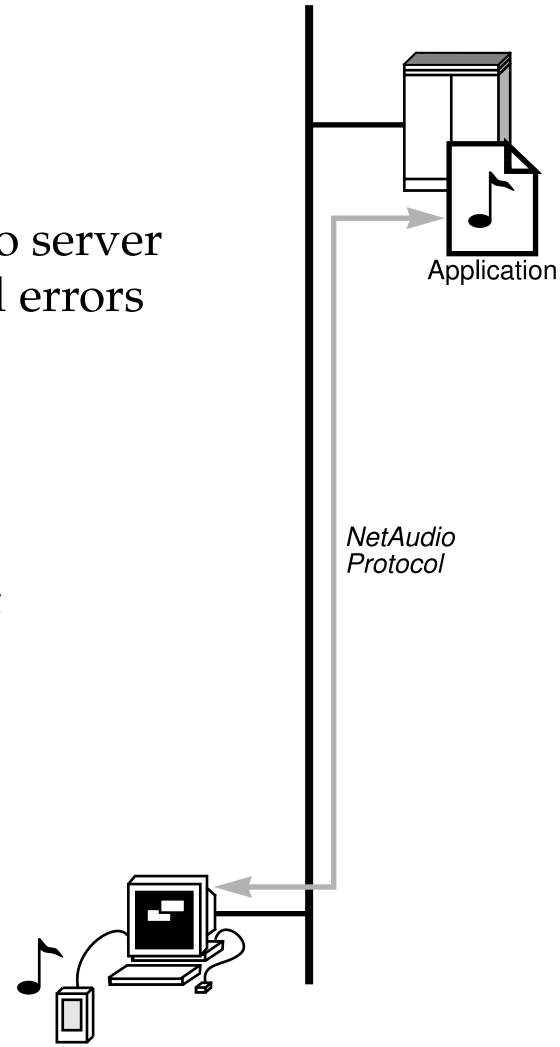
Manipulating Data Within A Flow

- Streams of values are added, multiplied, etc.
- Constants in operators can be changed dynamically
  - Allows for dynamic changes in gain
- Server can route data from inputs to outputs directly
  - Data only goes to clients when requested



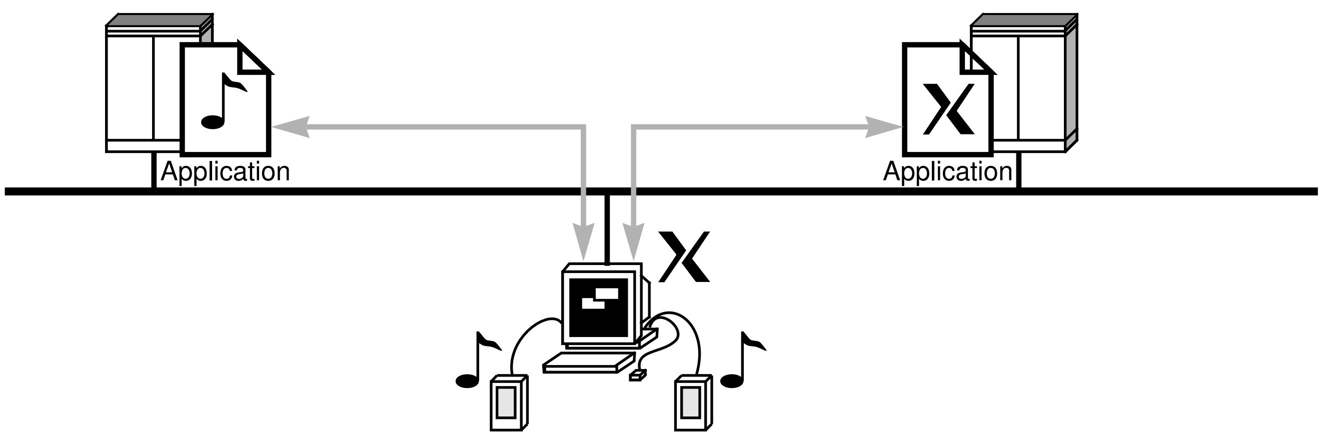
The NetAudio Protocol

- Similar to X Window System
  - Asynchronous, virtual stream connection to server
  - Client sends requests, gets back replies and errors
  - Server sends events
- Multiple byte-orders
  - Client sends in native byte-order
  - Audio data of either byte-order can be sent
  - Server handles any conversions
- Miscellaneous
  - Hooks for access control
  - Support for multiple versions and extensions



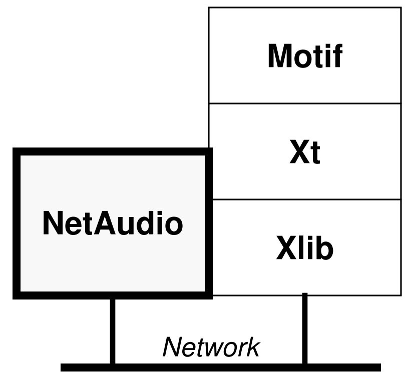
Synchronization

- Buckets (server-stored sounds) quick to play from
- Looking at integrating with X SYNC extension
  - Allow tighter synchronization with graphics
- Examining synchronization primitives in AudioFile



Application Programming Interface

- Low-level protocol wrapper layer
  - Equivalent to Xlib
  - Provides access to protocol
  - Routines can be synchronous (return status) or not (get errors)



- Higher-level utility layer
  - Convenience routines for common operations
  - Event handling framework
  - Simple interfaces to playing and recording **<u>many</u>** file formats
- X Toolkit interfaces
  - One call to register NetAudio connection with Xt
  - Adds Xt input callback to automatically deal with events
  - Sets audio library to automatically flush output

Summary: NetAudio Provides ...

- Distributed audio in heterogeneous client/server environments
- Portable, hardware-independent audio
- Support for many different audio data formats
- Use by a growing number of ISVs