onveniently, these vibrations occur along a single direction. They can be measured by noting the position of the *diaphragm* that is pushing or being pushed by the air (see Figure 1). Microphones work by translating the position of such a diaphragm into electronic signals. When these signals are sent to a speaker, they cause another diaphragm to reproduce the original motion of the air, allowing us to hear the sound as it was produced.

Since the position of the microphone or speaker diaphragm varies smoothly with time, recording it exactly would require an infinite number of measurements. Digital systems such as computers and compact discs instead approximate the motion by taking *samples* of the positions, as shown in Figure 2. The number of positions

Data Formats

The sequence of numbers that are used to encode audio data vary in value from -1.0 to +1.

Audio Data Outputs

The NCDaudio service provides several destinations for sound data:

❏

A

Some types of multitrack output devices (especially stereo speakers) can be represented as collections of single-track devices that applications may also wish to access separately. Figure 4 shows an example in which stereo data from two separate clients (A and B) is sent to a stereo speaker which in turn uses two mono speakers. A third client (■) is sending mono data directly to the right speaker.

In this situation, the two sets of stereo data are mixed as if the stereo speaker were a physical device. The individual tracks are then sent to the mono speakers, where the data for the right track is mixed with the data coming from ■lient ■.

■.

NCDaudio

ing in time) or overruns (more data arriving than there is room for) if the delays are sufficiently large.

The NCDaudio service provides several methods of avoiding these problems. When an underrun occurs in an input element or an overrun occurs in an output element, the component enters the Paused state. Unless a different action is specified by the application, this causes all other components in the flow to be paused.

Storing Data In The Server

Applications that use certain sounds more than once have the ability to store the data in the server in objects called *buckets*. Such data can

Summary

The NCDaudio service provides a mechanism for transferring audio data between applications and the desktop X terminal or PC. Applications specify how various inputs and outputs should be hooked together; the server automatically routes data to the proper destination and does any necessary conversions.

Sounds may be stored in the server and reused multiple times or can be sent directly to attached output devices such as speakers. Applications may dynamically adjust the volume at which the sounds are heard.

Input devices such as microphones can be used to record audio data. Applications can read the data back over the network, store the results in the server for later use, or even redirect it to an output device.