Chapter 7 Conclusion and Future Work

7.1. Conclusion

In this paper, we present a scheme for implementing a multiuser network audio system in UNIX-based operating systems. In our MuNAS, audio applications use the traditional open, ioctl, write, and close system calls to audio devices, rather than having to invoke a special audio library. Thus, the user can use legacy audio applications in our audio system without modification or recompilation. The MuNAS extends the X window system by providing network audio functionality. Our implementation of MuNAS in Linux shows that it is practical and provides good performance in an ordinary LAN (100 Mbps) environment.

7.2. Future Work

In our experimentation, MuNAS is very stable. In the future, we could improve MuNAS by embedding socket in kernel for shortening the time waiting acknowledgement and reducing the use of ack-receiving daemon. On the side, our schema is a suitable model for other character device, such as printer device /dev/lp.

7.2.1. Embed Socket in Kernel of Operating System

We have socket program to receive data from thin-client. We could make this daemon embed in kernel (Figure 6), i.e. it is kernel-level socket server that can handle
the packet transmitted from network. It is extra fast. So, we could estimate the cost
time of reading data relay. On the other hand, we don’t need to run another daemon.

![Diagram of system calls and processes](image)

**Figure 6**: Embed socket in kernel of operation system

### 7.2.2. Other Applications

Our Schema could also be applied for other area, such as printer. Sound device
belong to character device, and printer is too. We send audio data to `/dev/dsp`, and the
data that would be printed is sent to `/dev/lp`. Similarly, there are data relay, the
system-call serializer, and some daemons for send information via network. Then we
could also deal with printing document as playing sound.