

クロスプラットフォームなコンピュータネットワーク上での 音声通信の実現

篠田 晃 石井 一彦 桑名 栄二
{poripori, ishii, kuwana}@mickey.ntt.jp

NTTソフトウェア研究所
NTT

概要

最近、ワークステーションやパソコンにはオーディオ/ビジュアルデバイスが内蔵されマルチメディア情報を取り扱うことが可能となってきた。

一方、オフィスワークの進め方は、個人で仕事を進めるという形態よりもグループで協調的に作業を進める形態が多く、多くの組織で一般的になりつつあり、グループ内のコミュニケーションの支援が必要となってきた。また、オフィスでのネットワークには、種々のプラットフォームを持つコンピュータが混在しており、プラットフォームに依存しないクロスプラットフォームにおける統合的なマルチメディアコミュニケーション環境が重要となる。

今回、クロスプラットフォームからなるコンピュータネットワーク環境における音声通信ツールを試作した。本報告では、その構成法、性能、さらに試作から得た問題点について述べる。

Audio communication system on a cross-platform computer network environment

Akira Shinoda Kazuhiko Ishii Eiji Kuwana
{poripori, ishii, kuwana}@mickey.ntt.jp
NTT Software Laboratories
NTT

1-9-1 Kohnan Minato-ku, Tokyo 108 Japan

Abstract

Recently workstations (WSs) and personal computers (PCs) have audio/visual devices so that we can handle multimedia information easily. In an office environment, many office processes are done by teams rather than an individual work by utilizing these WSs and PCs. And in the office LAN, however, there are many kinds of computers and many kinds of platforms. Though it is widely accepted that supporting group communication for the computer based collaborative work is important, we think that an integrated multimedia communication environment on the cross-platform computer network is also important for the collaborative works.

In this paper, we describe an architecture of the audio communication system on the cross-platform computer network, its performance and related issues.

1. はじめに

近年、個人で仕事を進めるといった形態よりもチームやグループで協調的に作業を進める形態が、どの組織を見ても一般的になりつつある。協調作業を成功に導くためには、チームやチーム間のコミュニケーションの支援が重要である。例えば、Curtisは実際のソフト開発プロジェクトの分析から、開発担当者間、開発担当者と顧客間、開発担当者和その管理者などのコミュニケーションを取り上げ、言葉の意味の定義、設計内容の表記法の統一などを事前に行ない、担当者間のコミュニケーションチャネルを用意しておくことが重要であると述べている[Curtis88]。さらに、[Malone92]は、人間の知的作業に対して、コンピュータが処理できる形式情報の他に、人間が処理できるインフォーマルな情報を扱うことが重要であると述べている。例えば、マルチメディア情報は文字データ型の形式情報に付随するインフォーマルな情報を扱うことができ[桑名93]、コミュニケーション支援に有用な手段であると考えられる。

さて、近年のワークステーションやPCにはオーディオ/ビジュアルデバイスが内蔵されマルチメディア情報を取り扱うことが可能となってきた。これらのデバイスを利用することでインターネット上でのコミュニケーションに音声や映像情報を含んだマルチメディア情報を扱うことができる。例えば、UNIX環境において音声同期通信ツールとしてradio[Rossum92]、nevot [Schulzrinne92]などが実験的に提供されてきている。

一方、実際のオフィスには、UNIXワークステーション、Macintosh、Windows用のPCなど、種々のプラットフォームを持つコンピュータが混在しており、プラットフォームを制限すると相互通信ができないなどの問題が発生する。つまり、単一のプラットフォームではなく、クロスプラットフォームにおける統合的なマルチメディア

コミュニケーション環境構築法は1つの重要な研究課題と考える。

本研究では、クラスプラットフォームからなるコンピュータネットワーク環境における音声通信ツールを試作した。本ツールは、UNIXワークステーションとMacintoshを対象に、特殊なツールやデバイスを使うことなく、コンピュータに内蔵されているオーディオデバイスとシステムコールを用いて実現した。通信プロトコルとしては、TCP/IPプロトコル上のUDPパケットをもちいた。本報告では、本ツールを例にクロスプラットフォームにおける、音声通信環境の構成法、性能、さらに試作から得た問題点について述べる。

2. クロスプラットフォーム

半導体技術の進歩によりICチップは、ローコストハイパフォーマンスとなり、コンピュータも高性能のものが安く入手できるようになってきた。ネットワークインターフェースについてもオプションではなく、製品に組み込まれて出荷されるようになり、ネットワークへの接続も簡単に行えるようになってきた。

コンピュータの使用方法は、スタンドアロンからネットワークに接続しプリンターやディスク資源をグループで共用する方法に変わってきている。

ネットワークへ接続できるコンピュータの種類が増えてきて、LANのなかでさえもさまざまなコンピュータが接続されることもある。そしてプラットフォームもマシンに合わせさまざまなとなる。

そのようなさまざまなコンピュータのさまざまなプラットフォームがある環境においてコミュニケーションを行う場合、ある特定のプラットフォームに閉じているのは、特定内のコミュニケーションしかとれず不便である。そこでプラットフォームを超えるものとしてクロスプラットフォームなツールが必要となる。

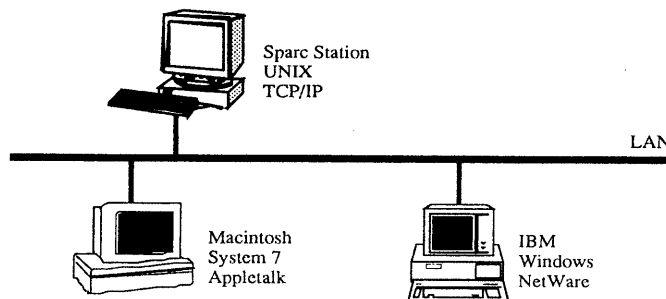


図1 クロスプラットフォームなインターネットの例

2. システム構成

今回の実験におけるシステム構成を図2に示す。

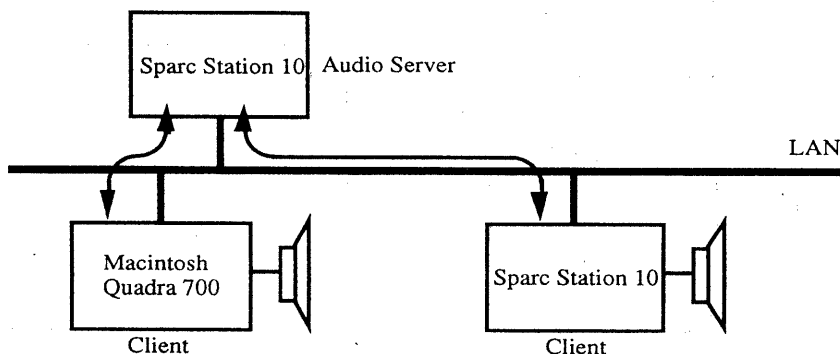


図2 システム構成

現在の対象コンピューターは、Sun Sparc Station 10 と Macintosh Quadra 700 を使用してクロスプラットフォームでの通信を可能としている。オーディオ信号の扱いは、それぞれのマシンに純正で搭載されているオーディオデバイスを使用している。

3. 利用形態と音質

音声通信の形態としては以下の3つがある。

- (1) ブロードキャスト
ネットワーク上の全マシンにデータを送る。この場合、1方向通信となる。
- (2) 1対1
ある特定の2つのマシン間でコミュニケーションをとる。
- (3) 1対多
複数マシン間で通信を行う。会議通信のようになる。

上記3つの利用形態では目的が変わるため求める音質も異なる。ブロードキャストでは、有線放送のように音楽を流すこともあるためハイファイも要求される。一方、1対1や1対多で人間対人間で通信をするときは、会話重視なのでハイファイのような音質は不要であり明瞭度の高い音質が要求される。

音声をデジタル処理する場合、音質を向上させようとするときとサンプリング周波数や精度ビット数を大きくするためデータ量が増加する。音声通信をネットワーク上で行う場合ネットワー

クの容量や処理方法の仕方によりデータ量が制限される。そのため音質にも限界がある。

4. クロスプラットフォームな通信環境の実現

クロスプラットフォームな通信を行うには、それぞれのプラットフォームで共通な仕様を利用するか、あるいは実装する必要がある。表1にSun Sparc Station 10 と Macintosh Quadra 700 のオーディオデバイスを含むプラットフォームの仕様を示す。

CPUやOSのレベルから仕様が異なるため、Sun Sparc なら LIKEN²¹⁾、Macintosh なら eXodus²²⁾ のような環境を載せるという方法もある。しかし、この方法は1つのマシンに2つの環境が混在し、使い勝手を損ねるため行わない。他の方法としては、(1) 各々のハードウェアで動作する共通OSを構築し、その上にアプリケーションを構築する方法、(2) 各々のオーディオデバイスはハードウェアおよびOSに依存しているため、通信プロトコルのレベルで統一し、各々にアプリケーションを構築する方法などがある。(1)の方法は理想形態であるが現実的でないため、本試作では(2)の方法を採用した。

²¹⁾ LIKENはXウィンドウ上でMacintosh環境を実現するツール。X celerated systems社製である。

²²⁾ eXodusはMacintosh上でXウィンドウ環境を実現するツール。White Pine社製である。

表1 プラットフォームの仕様

Machine	Sun Sparc Station 10				Macintosh Quadra 700			
CPU	Super SPARC				MC68030			
OS	UNIX				System 7			
Network I/F	TCP/IP				Appletalk			
Audio Specification	Sample Rate	Encoding	Precision	Channel	Sample Rate	Encoding	Precision	Channel
	8000 Hz	μ -law	8 bit	1	0 Hz	linear	8 bit	1~4
	8000 Hz	A-law	8 bit	1	~ 44100 Hz			
	8000 Hz	linear	16 bit	1 or 2				
	9600 Hz	linear	16 bit	1 or 2				
	11025 Hz	linear	16 bit	1 or 2				
	16000 Hz	linear	16 bit	1 or 2				
	18900 Hz	linear	16 bit	1 or 2				
	22050 Hz	linear	16 bit	1 or 2				
	32000 Hz	linear	16 bit	1 or 2				
37800 Hz	linear	16 bit	1 or 2					
44100 Hz	linear	16 bit	1 or 2					
48000 Hz	linear	16 bit	1 or 2					

そこで、通信を行うために最低限必要なネットワークインターフェースと伝送データについて共通点を考える。

①ネットワークインターフェース

Sun Sparc Station 10 は TCP/IP、Macintosh Quadra 700 は Appletalk (Ethernalk) であるが Macintosh は、MacTCP により TCP/IP にアクセスできる。

②サンプリング周波数

Sun Sparc Station 10 は連続ではないが、Macintosh と共通な値を持つ。

③符号化方式

linear が共通である。

④精度ビット数

8 ビットが共通であるが、Sun は μ -law となる。そのため、linear の場合は、Macintosh 側で 16 ビットを 8 ビットに変換する必要がある。また、8 ビットの場合は、送信側か受信側で μ -law と linear の変換をする必要がある。

⑤チャンネル数

2 チャンネルまでは共通である。

①~⑤より以下の条件を設定しツールの作成を行った。

- ・ネットワークインターフェースは、TCP/IP とし、Macintosh では MacTCP を使用する。
- ・サンプリング周波数は Sun に合わせる。
- ・符号化方式に μ -Law を使用する場合は、Macintosh 側で linear に変換する。
- ・符号化方式に linear を使用する場合は Sun からのデータは 16 bit となるため Macintosh 側で 8 bit に変換する。
- ・チャンネル数は、1 チャンネルと 2 チャンネルの選択とする。
- ・Macintosh からのデータは 8 bit であるため、

Sun で 16 bit に変換する。

Sun からのデータを Macintosh 側での受信時に変換するのは、Sun では精度ビット数を 16 bit で扱い、音質的に Macintosh より上位であるためである。

4. 1 伝送方法

インターネット上の通信は、コネクション型の TCP とコネクションレス型の UDP がある。今回プロトコルは UDP を使用した。これは

- ①プロトコルがシンプルであること。
- ②パケットの欠落等で音がとぎれても聴感上補正できるため再送等のデータ保証が不要なこと。
- ③ブロードキャストが可能であること。

という点に着目した。

また、UDP は、サービスポート番号によりパケットを選ぶことができる、本ツールでは、これを利用してサービスポート番号によりオーディオの通信回線を識別している。

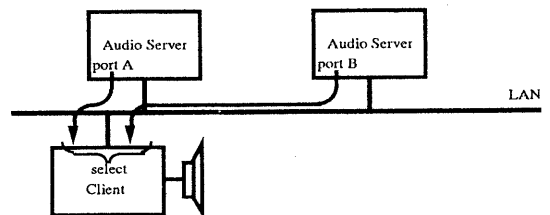


図3 通信回線の識別

転送データについては、送信および受信側の

負荷を軽減するためデータ圧縮はせず、オーディオデバイス出力からのロウデータとしている。

4. 2 Sun Sparc Station 10 での処理

(1) 送信時 (図4)

/dev/audio または、オーディオフォーマットファイルから read を使用してデータを読み込み、sendto でインターネット上へ送り出す。現在はロウデータであり、圧縮はしていない。

(2) 受信時 (図5)

recvfrom を使用してインターネット上からデータを読み込み、write により /dev/audio へ書き込む。

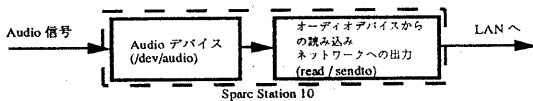


図4 送信時の処理

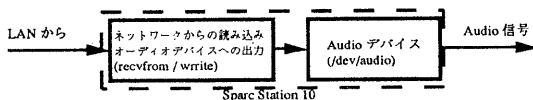


図5 受信時の処理

4. 3 Macintosh Quadra 700 での処理

(1) 送信時 (図6)

サウンドツールの出力を一旦ファイルへ取り込み、そのファイルのデータを MacTCP パッケージを利用し送信する。

(2) 受信時 (図7)

- ① MacTCP パッケージを利用して UDP パケットを受信する。
- ② 受信したパケットを内部バッファに格納する。Sun Sparc Station から送られるデータは (A) μ -law フォーマット、(B) 16bit リニアのどちらかであるのに対し、Macintosh で再生可能なデータは 8bit リニアなので変換を行い、バッファに格納する。
- ③ バッファがいっぱいになった時点で Macintosh のオーディオデバイスの制御を行っているサウンドマネージャーにデータを送る。

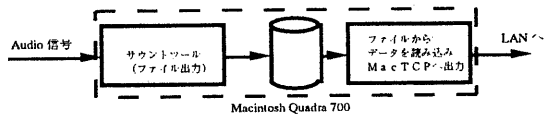


図6 送信時の処理

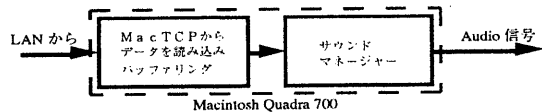


図7 受信時の処理

5. 評価

本章では①クロスプラットフォームな通信環境の実現性と②音質の2つの側面から評価を行う。

5. 1 クロスプラットフォームの音声通信サービスの評価

3節では、音声通信形態には、(1)ブロードキャスト型、(2)1対1型、(3)1対多型の通信があることを示し、4節ではそれらのサービスを実現するためのシステム構成について示した。しかし、表2に示すように、実際には、MacintoshなどのOS環境では(2)(3)のサービスの実現は難しい。

表2 音声通信形態とプラットフォームの関係

	ブロードキャスト	1対1	1対多
UNIX (例: SunOS)	○	○	○
Macintosh	○	△	△

凡例: ○ 特に問題なく実現可能、
△ 実現可能であるが実用上の問題がある

これは4.3節に示したように、Macintosh OS においては、以下のような理由によるものである。

- ・ライブラリの制約から音声取り込み中に取り込み以外の処理 (LAN への出力等) ができない。
- ・送信時に音声取り込みと並行に LAN への出力処理が可能になったとしても、オーディオデバイスは送受信時に共用しなければならない^{脚)}。

^{脚)} Sunでは、サンプリング周波数は同一であるが、1つのオーディオデバイスを論理的に別々に見せることができる。

つまり、送受信が同時に存在するリアルタイム型の音声通信サービスには、オーディオデバイスの共用は性能に大きな影響を与えるため現実的ではない。

5.2 オーディオデータの再現性の評価

5.1節では、クラスプラットフォームにおける音声通信サービスの実用上の問題を考察した。上述のように、Macintoshなどの疑似マルチタスクOS上では1対1および1対多の音声通信サービスにおいては、実用上の問題がある。

一方、実用上の問題とは別に、プラットフォームとは関係なく、コンピュータネットワークにおけるデジタル音声通信の性能という評価項目が存在する。そこでこの章では、デジタルオーディオデータの再現性を確認する。

デジタルオーディオは、サンプリング周波数と精度ビット数を上げることにより音質が向上する。しかし、その反面データ量が増加する。さらに、チャンネル数の増加によってもデータ量が増加する。データ量の増加によりネットワークでの転送時間とコンピューターでの処理時間が増加し、オーディオデータを再現できなくなることが予測できる。

そこで、サンプリング条件と転送データ長を変更して再現性を確認した。確認を行った環境は、送信側、受信側それぞれに Sun Sparc station 10 を使用しネットワーク上は2台以外のマシンを接続しないで行った(図8)。結果を表3に示す。

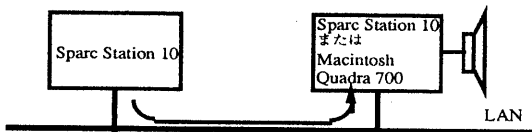


図8 実験環境

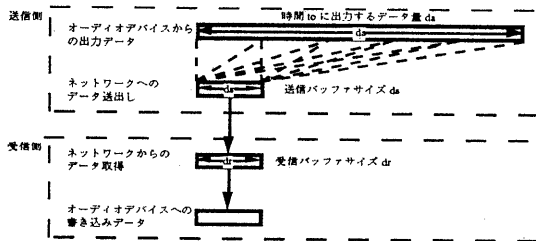


図9 バッファリングの状態

次に処理時間について考える。

時間 t_0 に出力されるデータ量 d_a は、サンプリング周波数 f_s 、精度ビット数 bit 、チャンネル

表3 オーディオデータの再現性

Sample rate (Hz)	Precision (bit)	Encoding	マシン	
			Sun SPARC station 10	Macintosh Quadra 700
8000	8	μ -law	◎	◎
8000	16	linear	◎	◎
9600	16	linear	◎	◎
11025	16	linear	◎	◎
16000	16	linear	◎	◎
18900	16	linear	◎	○
22050	16	linear	◎	○
32000	16	linear	◎	○
37800	16	linear	◎	×
44100	16	linear	◎	×
48000	16	linear	◎	×

データサイズ = 1400 byte

凡例 → ◎ : ステレオ再生も可

○ : モノラルのみ再生 × : 再生失敗

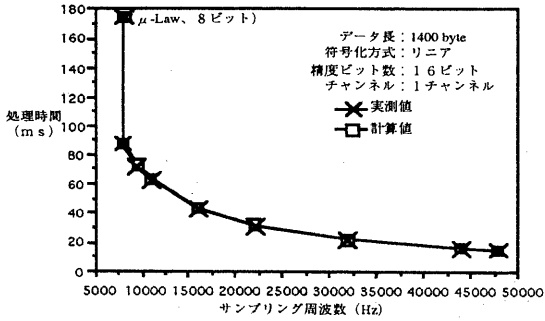
ル数 ch とすると (図9)

$$d_a = f_s \times \frac{bit}{8} \times ch \times t_0 \dots (1)$$

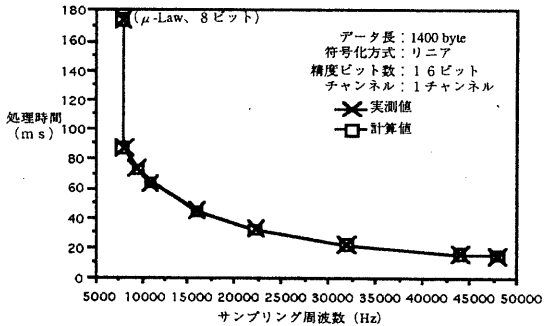
である。送信バッファサイズを d_s 、受信バッファサイズを d_r とし、 $d_s = d_r$ とすると、送信データを受信側で再生するのに必要な時間 t_p は、

$$\begin{aligned} t_p &= \frac{d_s}{d_a} t_0 \\ &= \frac{d_s}{f_s \times \frac{bit}{8} \times ch \times t_0} t_0 \\ &= \frac{d_s \times 8}{f_s \times bit \times ch} \dots (2) \end{aligned}$$

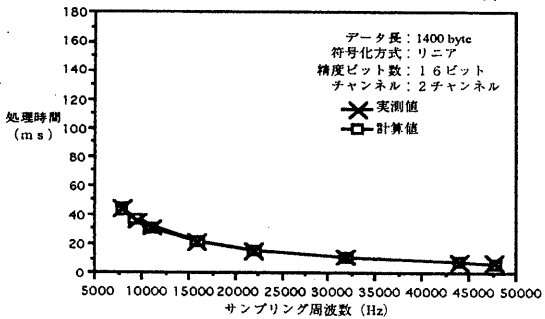
となる。つまり、サンプリング周波数 f_s 、精度ビット数 bit 、チャンネル数 ch の値を大きくすると送受信バッファサイズが一定ならば t_p はそれらの値に反比例して小さくなるため受信側での処理を高速にする必要がある。また、処理速度が一定ならば、サンプリング周波数 f_s 、精度ビット数 bit 、チャンネル数 ch の値の変化に合わせ、処理速度が追従するように送受信バッファサイズを変化させる必要がある。これを確認するため、各処理時間を測定した。この結果を図10に示す。(環境は図8である。)



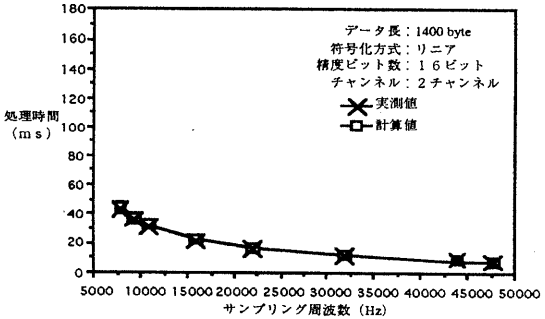
受信側の処理時間 (1チャンネル時)



送信側の処理時間 (1チャンネル時)



受信側の処理時間 (2チャンネル時)



送信側の処理時間 (2チャンネル時)

図10 処理時間の測定結果

□は計算値である。図10中のどの結果も受

信側処理と送信側処理で、これは表1の結果と同じであり、式(2)のとおり処理が行われていることを示している。

ところで、Macintoshでは、擬似的にマルチタスクを行っているためタスクスイッチによりアプリケーションの動作を切り替えている。ウィンドウ上のアプリケーションをいつでもすぐに使用できるようにアプリケーションに割り当てられる処理時間に制限がある。この制限はOSからのもので1/60秒となっている。したがって、この値より高速にできないので、図10での測定で使用したデータ長(1400byte)では、リニアの符号化方式、精度ビット数16bitでチャンネル数が1ならサンプリング周波数37800Hz、チャンネル数が2ならサンプリング周波数18900Hzのデータまでしか再現できない。表3でMacintoshが再生を失敗しているのは、そのためである。

つまり、Macintoshで音質を向上させるためにサンプリング周波数とチャンネル数を増やすなら、式(2)に従い、送受信バッファサイズ(データ長)を大きくする必要がある。

デジタルオーディオデータの再生のポイントは処理時間であり、データ再生時間が再生データサイズに従う処理時間に追従するというのが必要である。

今回はオーディオデバイスからの出力データを転送し再生を行っている。オーディオデバイスからの出力データはある一定周期でサンプリング周波数、精度ビット数、チャンネル数に従い出力されるデータであり、デバイスからの出力時に既に周期的に出力される。一方、オーディオデータファイルからデータを取得し転送、再生場合は、送出側で式(2)に従いデータを周期的に送出する必要がある。

送受信側の処理は、基本的にはデバイスからデータを取得しネットワークへ送る(受信時はネットワークからデータを取得しデバイスへ送る)だけである。しかし、実際に処理を構築する場合は、エラー処理やイベント処理も必要であり、オーディオデータの処理だけではなく、そのため実際には、処理時間も式(2)で計算する時間よりも多くかかり、処理の設計をする場合もオーディオデータ処理以外の処理時間を考慮する必要がある。

試作では計算値より1つ低い値でのデータ再生になることが多かった。

6. おわりに

本報告では、クロスプラットフォームなコンピュータから構成されるネットワーク上での音声通信システムの構成法について示した。具体的には、

- ・リアルタイムデータの処理に際しプラットフォーム上でのプロセス制御法、
- ・処理アルゴリズム

を示した。

また、実際にクラスプラットフォーム環境として、Sun SPARC station 10 と Macintosh Quadra 700 を使用した音声通信システムを試作し、その実現可能性について検証した。

試作後、処理を追って実際の処理の確認、処理時間の測定などの性能実験を通して、デジタルオーディオデータ伝送時の再現実験とそのデータから得られた以下に示す実現上の問題点/注意点について述べた。

- ・同期的な通信には現在のOSではプロセス制御とネットワークインターフェースの処理速度に限界がありリアルタイム性が必要である。特にシングルタスクを擬似的にマルチタスクとして動作させている場合は適さない。
- ・デジタルオーディオデータの伝送/再生にはサンプリング条件に合わせて処理速度を追従させる必要がある。また、処理速度が一定ならば転送データサイズを追従させる必要がある。

今後、今回の試作、実験に引き続き以下を行う予定である。

- ・伝送遅延に対する実験：LANのみならず、ネットワーク機器を介したWAN環境における伝送遅延に関する実験を行い、さらなる音声通信の実用性について検証する。
- ・TCP、ST-II [Topolcic90]を使用した通信実験：グループ通信や放送型の通信など種々のコミュニケーションサービスと通信プロトコルの適合性を検証する。
- ・インターネット以外での音声通信サービスの実現可能性の検討：パソコンで主流となっている NetWareなどの通信プロトコルを用いた音声通信サービスのシステム構成法とその可能性について検討する。

謝辞

本検討にあたり、NTTソフトウェア研究所ソフトウェア開発技術研究部 中村主幹研究員、坂本泰久氏、同ソフトウェア基礎技術研究部 小川主幹研究員 さらには分散ソフトウェア開発方式研究グループの諸氏から貴重なコメント及びご指導を頂いた。記して感謝する。

【参考文献】

- [Curtis88] Curtis, B., et al. "A field study of the software design process for large systems." C. ACM., 31, 1988.
- [Malone92] Malone, T., Lai, K., Toward Intelligent Tools for Information Sharing and Collaboration, Computer Augmented Teamwork (Ed. Bostrom, et al.), Van Nostrand Reinhold, 1992.
- [桑名93] 桑名, 増尾., "マルチメディアコミュニケーションが意味レベルで人をつなぐ," ビジネスコミュニケーション, Vol. 30, March/April 1993.
- [Rossum92] Guido van Rossum, "Radio," Stichting Mathematisch Centrum, Amsterdam, The Netherlands, 1992.
- [Schulzrinne92] Henning Schulzrinne, "Voice Communication Across the Internet: A Network Voice Terminal," University of Massachusetts, July 1992.
- [Topolcic90] C. Topolcic, "ST II," in First International Workshop on Network and Operating System Support for Digital Audio and Video, no. TR-90-062 in ICSI Technical Reports, (Berkeley, CA), 1990.