

オンデマンドオーディオサーバーの実現

分散オーディオ/ビジュアルサーバーの提案

篠田 晃 石井一彦 桑名栄二*

*NTTソフトウェア研究所
NTT

情報の提供形態が静的な情報から動的な形式で提供される形態（例えばマルチメディア情報提供）へ変化している。今回我々は、マルチメディア情報提供サービスへ向けた実験として、一般になりつつあるTCP/IP上でのオーディオ情報提供サービスを実現した。オーディオサーバーはUNIX (Sun)、クライアントはクロスプラットフォームをターゲットとして UNIX (Sun) と Macintoshとする環境でシステムを実現した。本論文では、分散オーディオ/ビデオサーバーの実現方式、サービスの実現方式、および実際の利用経験から得た今後の課題について述べる。

1. はじめに

情報の取得形態としては受動的な方法と能動的な方法がある。例えば、テレビ、ラジオはスイッチを入れておけば情報が流れてくる。また、契約さえしておけば、新聞は配達してくれる。

一方、NTTの104番やダイヤルQ2は情報を得たい人がサービスへアクセスして情報を取得する。

インターネットにおいては、メーリングリストによるプリテンボードが受動的な情報収集であり、それ以外は自ら能動的に情報を収集していく場合が多い。コンピュータネットワークにおいては、情報共有という考え方が強い。つまり、「○○の情報には××に△△というファイルになっているから、それを見てもいい」というようにである。このような考え方から、情報は、あるマシンのあるディレクトリに置き、アクセスは、rlogin、telnet、rcp、ftp等で行う、あるいは近隣のマシンであれば NFS でリモートマウントしてアクセスするという方法をとってきた。最近では、gopher[1]、wais[2]、WWW[3]のようなクライアント/サーバータイプの情報アクセスツールを使用し、容易に情報アクセスができるようになってきている。また、ネットワークが拡張され情報の存在エリアがネットワークワイドになり、複数の情報提供サーバーに存在

する多くの情報を効率よくブラウジングしながら情報を取得するという傾向にもなってきている。

提供情報は、静的な情報（例えば個人情報、資料名情報）が多い。WWWはハイパーメディアによる情報提供を目的としているが、現在ではまだ静的な情報が中心である。しかし、今後は静的な情報提供から情報アクセス時に情報がアプリケーションによりリアルタイムに処理され、動的な形式で提供される形態（例えばマルチメディア情報提供）が多くなると考えられる。

本研究では、一般になりつつあるTCP/IP上でのオーディオ情報提供サービスを例として、コンピュータネットワーク上での分散オンデマンド型マルチメディアディレクトリサービス環境(PowerDirectory) 構築方式を検討した。オーディオサーバーはUNIX(Sun)、データはISDNで使用しているμ-LawやCD等で使用しているリニアのオーディオデータを使用し、クライアントはクロスプラットフォーム[4]をターゲットとしてUNIX(Sun)とMacintoshとする環境でシステムを実現した。本論文では、分散サーバーの実現方式、応用例、および実際の利用経験から得た今後の課題について述べる。

An experimental of on-demand audio server
Akira Shinoda, Kazuhiko Ishii, Eiji Kuwana *
* NTT Software Laboratories, NTT

2. パワーディレクトリ

ディレクトリサービスを規定している X.500 はパーソナルな情報に対するアクセスである。情報は1個人毎にまとめられており、名前や電話番号等の項目は1個人に対する属性として位置づけられている。

ディレクトリサービスがパーソナル情報の提供であることに対して、一般的なもの(オブジェクト)に対する情報提供サービスを我々はパワーディレクトリサービスと呼ぶことにした。ここで言うオブジェクトとは、椅子、机、車のように物理的にも存在するものから映像、音楽、プログラムのようなソフトウェア(サービス)的なものまで、物として認識できるものとする。

パワーディレクトリは、複数のオブジェクトを持つことができる。オブジェクトは抽象度の高い物である。例えば、車、映画、音楽のようなレベルである。オブジェクトは、オブジェクトを特徴付ける属性を持つ。オブジェクトの具体的な物はアイテムとなる。

例えば、オブジェクト=車は、属性=エンジン、タイヤ、サスペンションを持っており、アイテムとしてはシルビア、プレリウド、セリカとして分類する。

PowerDirectory - Object - Item - attribute

図1 データ構造

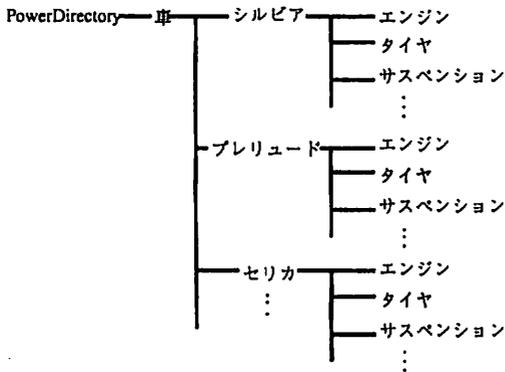


図2 例

車、ステレオ、テレビ等の物理的なオブジェクトは、コンピューターネットワーク上ではオブジェクト自体を提供することはできず、アイテム毎の属性を静的な情報としてしか提供できない。

一方、音楽、ビデオ、映画等のオブジェクト

は情報であり、コンピューターネットワーク上をコンピューターデータとして流通させることが可能である。そこで、パワーディレクトリでは、音楽、映像さらにマルチメディアを含めた動的な情報も提供する。

3. 分散サーバーによる情報提供

情報提供を行うサーバーは、できる限り多くの情報を蓄積しようとするが、アクセス集中に対する負荷の低減、情報の分類と蓄積量の軽量化のためサーバーを分散させることも必要となる。また、メディア毎に分類することによりサービスの幅も広がる。

今、オーディオサーバー、ビジュアルサーバー、インフォメーションサーバーがコンピューターネットワーク上に存在することとする。これらサーバーは単独でアクセスも可能である。

さらにムービーサーバーがあり、ある人が映画を見るためにムービーサーバーにアクセスすることとする。映画は、特に日本向けの洋画の場合は、映像、音楽に加え字幕というテキスト情報があり、マルチメディア情報となっている。

ムービーサーバーは、ストーリー展開をタイムスケジュールとして保持しており、タイムスケジュールに従い、オーディオサーバー、ビジュアルサーバー、インフォメーションサーバーの3つのサーバーへ情報の提供を指示する。

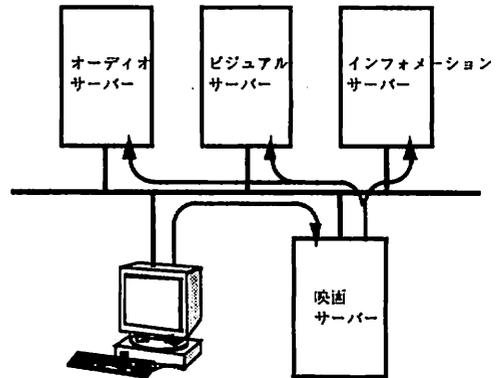


図3 クライアントからのアクセス

ユーザーはクライアントでムービーサーバーを通して3つのサーバーからの情報を組み合わせて1つの映画を見ることができる。

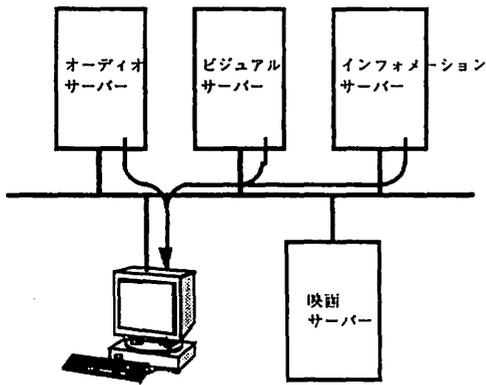


図4 クライアントへの情報提供

インフォメーションサーバーには映画の原作があるとする。ユーザーは、原作を読むために、インフォメーションサーバーにアクセスする。ここでも原作のストーリー中の場面に合わせて、オーディオサーバーとビジュアルサーバーにアクセスポインターが張ってある。そのためユーザーは原作を読みながら、サウンドトラックを聴いたり、情景を見たりすることができる。

音楽にも詩があり、想像する情景がある。これを考慮し、オーディオサーバーもビジュアルサーバーやインフォメーションサーバーにポインターが張ってあり、音楽を聴きながら歌詞や映像を見たりすることが可能となる。

このように中心になるメディアにより、他のメディアの組み合わせ方を変えることができるため、メディアごとにサーバーを分散させることにより多くの利用形態が可能となる。

最近では、データの配布の高速化と省力化を目的としてISDNを使ったカラオケシステムがデータを分散させている。この場合の分散は

オーディオデータと歌詞のみをリモートのサーバーから転送する。ビジュアルデータはローカル側に蓄積されたものを使用する。この形態は本章で述べている分散サーバーの一形態である。

4. オーディオサーバー

今回はマルチメディア情報提供の一例として、オーディオサーバーを取り上げ試作/実験を行った。この章では、試作したオーディオサーバーについて述べる。

4. 1 実現環境

サーバーはUNIXマシン (Sun) 上とし、クライアントとはTCP/IPにより通信を行う。オーディオデータ用のストレージとしては、カセットテープ、CD、DAT、MD、DCCが一般的であるが、それらのストレージをドライブするためのハードウェアとUNIXマシンのインターフェースは多くない。また、今回の条件として次の3つが必要である。

- ①アクセスが高速である
- ②読み書き可能である
- ③同時アクセスが可能である

これらの条件は前述の5つのストレージはどれも満足しない。特に、同時アクセスが行えるドライブはないため、サーバーはクライアントからの複数同時アクセスに対処するためには、専用のハードウェアが必要となる。

そのため、特殊なハードウェアを使用しないことも含め、UNIXファイルシステムを使用することとした。

クライアントは、現在、UNIX (Sun) と Macintosh 上で実現している。

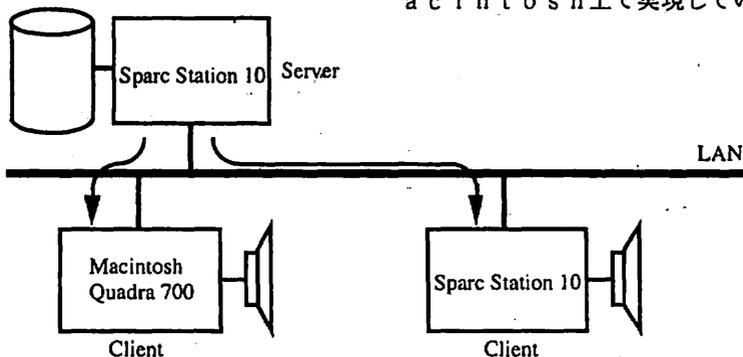


図5 実現環境

4. 2 サーバー

／クライアント間の通信

サーバー／クライアント間の通信は、ユーザーの行動をモデル化した動作より必要な操作ををコマンドとして行うようにした。

ユーザーの行動のモデル化

ユーザーがサーバー（UNIXマシン）から音楽（音）を聴き終るまでの行動を考えた。ユーザーの目的は音楽（音）を聴くことである。そのため、次の行動を基本とした。

- ①サーバーへ接続する。
- ②ファイルを選択する。
- ③ファイルをプレイする。
- ④サーバーとの接続を切断する。

これらに加えて、以下の条件の場合の行動が付加される。

- ・希望のファイルが見つからなかった場合
- ・プレイを止めたい場合
- ・プレイを一時的に止めたい場合

これらの条件の場合の行動は

- ⑤ファイルを探す→ディレクトリを移る。
- ⑥プレイを止める
- ⑦プレイの一時停止
- ⑧一時停止の解除

となる。

1～8の行動を時間的な変化を考慮して記述すると図6に示す状態遷移図となる。

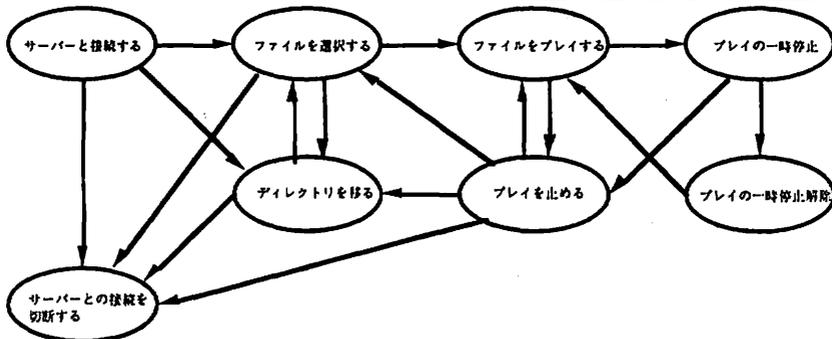


図6 ユーザー操作の状態遷移図

図6をもとにオーディオデバイスに必要な動作を含め、表1に示すコマンドセットを作成しサーバー／クライアント間の通信を行うことと

した。

表1 クライアントからのコマンドセット

コマンド	意味
Change Directory	ディレクトリを移る
Get List	ファイルリストを得る
Select File	コンフィグレーション情報を得る
Play File	オーディオデータを再生する
Stop Playing	オーディオデータ再生を中止する
Pause Playing	オーディオデータ再生を一時停止する
Release Pausing	再生の一時停止を解除する

4. 3 オーディオ情報データベース

必要なオーディオデータを選択するためには、そのデータに関する情報を提供することが必要である。

SunOS では、オーディオデータファイルにヘッダーを付けることができ、そのヘッダー内に情報を格納できる。格納情報を表2に示す。

表2 オーディオファイルヘッダー情報

情報
サンプリング周波数
ファイル内のサンプリング周波数の数
ファイル内の精度バイト数の総和
チャンネル数
符号化方式
データサイズ

これら情報ではオーディオデータ再生時に必要なコンフィグレーション情報は得られるものの、ユーザーが選択するためのインフォメーション

ョンとして付加できる情報は少ない。

そのため、今回はデータベースを使用し、オーディオデータファイルのヘッダー情報とあわせてクライアントへ提供することとした。

データベースは特殊なツールは使用せず、テキストファイルとした。

これをオーディオデータファイルのあるディレクトリに置くことにより使用する。

データベースの格納情報は以下のものとしている。

表3 データベースの格納情報

情報	データ内容
ファイル名	ファイル名
ファイル種別	ディレクトリとファイルの区別
タイトル	録音内容等
アーティスト名	録音した人、録音された人の名前
再生時間	再生時間
録音日付	サーバーに置かれた時間

4.4 伝送方法

4層でのプロトコルはコネクション型のTCPを使用した。これは

- ・サーバー/クライアント間は1対1の通信であること。
- ・サーバーはクライアントとの複数接続を行うため、コネクションの管理がUDPより簡単であること。

という理由のためである。

また、コネクションは2コネクションとし、コントロール用とオーディオ用にそれぞれ使用している。

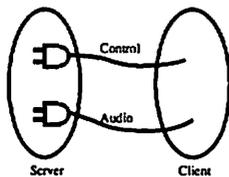


図7 TCPコネクション

4.5 データ転送方法

動的な形式での情報提供であり、情報アクセス時に情報のリアルタイム処理を行う。そのため、オーディオデータ自体を転送しクライアン

トでのリアルタイム再生を行う。

転送データはファイルから読み込んで転送する。データはサンプリングされているので、サンプリング周波数への同期が必要である。そのため、送信側で転送データ量とサンプリング周波数に従い周期的にパケットを送出する。受信側は送信側のパケット送出タイミングでパケットを読み込むので、サンプリング周波数に同期をした再生が行える。

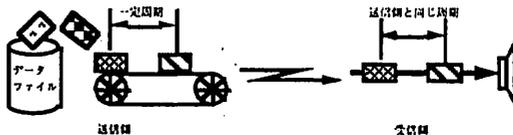


図8 同期のとり方

4.6 オーディオデータ形式

サーバーからのデータ送出は、単純にクライアントへデータを送るだけであり、データ再生に影響する要素がない。ところで、今回のサーバーの実現環境はSunであり、クライアントのオーディオコンフィグレーション情報はSunのオーディオライブラリを使用してファイルヘッダーより取得している。そのため、オーディオファイルヘッダーがついたSunのファイル形式である必要がある。

オーディオデータのサンプリング周波数、符号化方式等はクライアントで再生できるものであればよいため特定しない。ただし、サーバー側で圧縮データに対する対応は行っていないため、ロウデータであることが必要である。

5. 応用例

今回、作成したシステムを使用した応用例を考える。

(1) オーディオサーバー

これは今回作成したシステムの当初の目的であり、クライアントからのアクセスにより聴きたい音楽を聴くというシステムである。

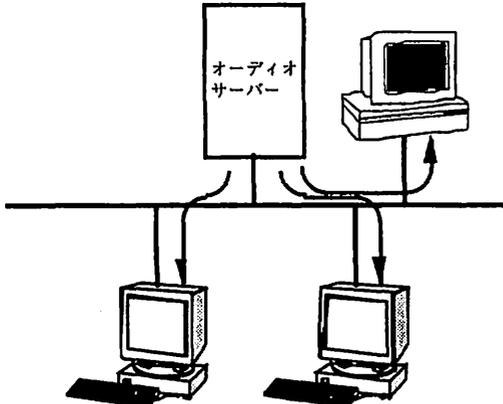


図9 オーディオサーバー

(2) 音声ブリンテンボード

サーバーに複数アクセスができるという機能を利用することにより、ブリンテンボードとして利用することができる。

例えば、ある人がメッセージとしてサーバーにオーディオデータファイルを置いておけば、それを多くの人がアクセスしてメッセージを聴けばよいのである。

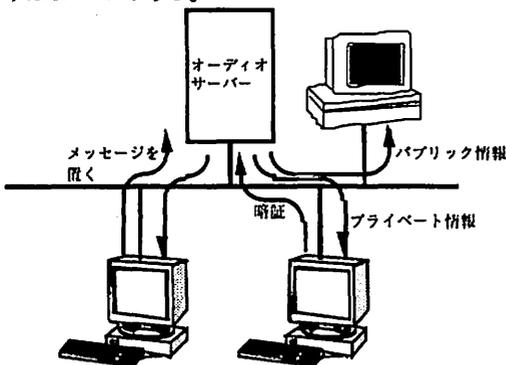


図10 ブリンテンボード/メッセージボード

(3) メッセージボード

ブリンテンボードは利容者全員に対してアクセスを許すが、個人に対するセキュリティを設けることにより「伝言ダイヤル」のようなメッセ

ージボードとして利用できる。

(4) アクセスユーザー間コミュニケーション

今回のようなツールを利用する場合、アクセスする人はその情報に興味のある人であり、ある目的を持った人である。今、Aという情報に2人の人がアクセスするとする。この2人はAという情報について共通の興味を持っている。そのため、Aという情報を話題にしてコミュニケーションをとることもできる。

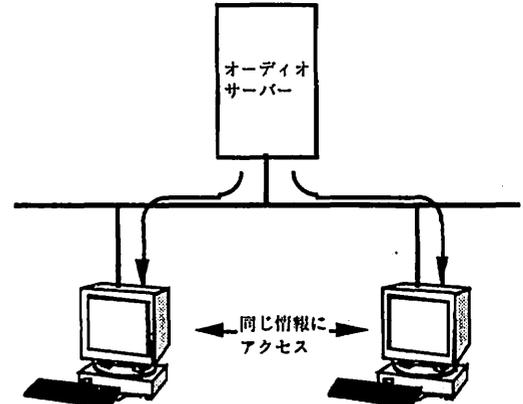


図11 同じ情報へのアクセス

そこで、今回作成したサーバーにある情報にアクセスした場合に、その情報にアクセスしている人を認識させる機能とその人と通信する機能を取り入れることにより一人で情報を利用している状態から、2人あるいはグループでの利用へ広がる。

今回のサーバーはオーディオの提供であるが、例えばゲームの場合であれば、サーバーにアクセスし、既にゲームを行っている人がいれば、2人で対戦ができる。

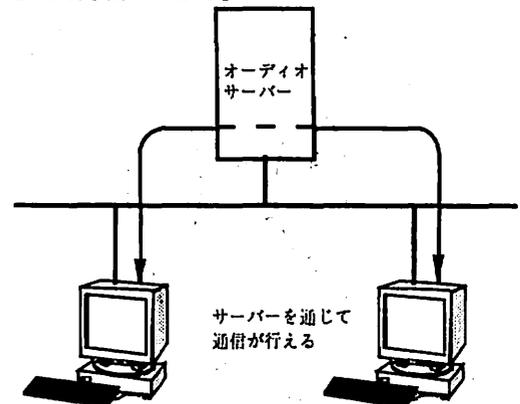


図12 サーバーを通じた通信

6. 今後の課題

6.1 ストレージの容量

オーディオのクオリティーを高くする場合、サーバーとして必要とするストレージの容量が大量に必要となる。例えば128MBのMOディスクへCDと同等の品質で蓄積するならば約12分しか蓄積できない。大容量のストレージを必要としなくても多くの情報を提供できるようにMPEG等の圧縮を取り入れる必要がある。

6.2 データ伝送時の品質の確保

今回のサーバー/クライアント間のデータ伝送では、サーバーからの送出時にオーディオデータのサンプリング周期に送出タイミングを合わせず方法をとった。しかし、この方法では伝送途中での遅延、サーバー側でのわずかなタイミングのずれが、クライアント側での受信時の途切れとなる。これを防止するためクライアント側での受信時にサンプリング周期に再生タイミングを合わせるという方法が考えられる。

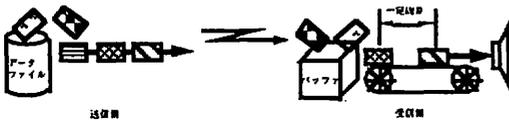


図13 受信側で同期をとる方法

具体的に次ようになる。

送信側：

データを送出するのみである。

受信側：

データを受け取り、サンプリング周期に同期させて再生を行う。

実際には、再生スピードよりデータ転送スピードの方がはるかに速いため受信側でのバッファのオーバーフローが考えられる。今回は4層のプロトコルにTCPを使用しているため、アプリケーション層のデータオーバーフローをTCP層で、どの程度カバーできるかも考慮する必要がある。

6.3 アクセストラフィック

試作したオーディオサーバーを利用する場合、オーディオデータ転送のトラフィックが発生する。

トラフィックは1アクセス当たり、オーディオデータ形式により表に示す量となる。

表4 トラフィックの計算

sampling rate	precision	encoding	channel	traffic
8kHz	8bit	μ -Law	Mono	64kpbs
44.1kHz	16bit	Linear	Stereo	1.4Mbps
48kHz	16bit	Linear	Stereo	1.5Mbps

表4の上から品質は、ISDN電話、CD、DATとそれぞれ同等である。

FDDI等の高速ネットワークの利用も考えられるが、Ether-netの利用が多いのが現実である。Ether-netの場合は、トラフィックが5~3Mbpsぐらいまでが快適なネットワークの状態である。表よりISDN電話品質であれば、30ユーザー程度の同時アクセスが可能である。しかし、CDやDATの品質を求めるなら、2ユーザー程度のアクセスしか可能でない。

トラフィックの緩和のため、データ転送時には圧縮データの転送も必要となるだろう。

6.4 分散サーバー間の連携

3章で述べたような分散サーバーを実現するため、他サーバーとの連携方法が必要となる。

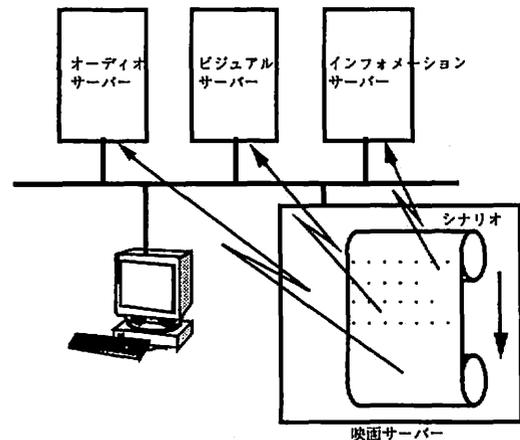


図14 分散サーバーの連携

連携時に必要なこととしては次の2つである。

- ①主体となるサーバーと連携をとるサーバー間のリンク
- ②メディア間の同期

図14に示すような映画サーバーを考える。映画サーバーは、3つのサーバー（オーディオ

オサーバー、ビジュアル、インフォメーションの各サーバー)ヘクライアントへの情報提供を指示する。

3つのメディア(オーディオ、ビジュアル、インフォメーション)は同期する必要があるが、同期は映画サーバーによって調整を行う。つまり、映画サーバーは3つのサーバーの同期情報を持っており、その情報によって3つのサーバーに指示を出すようにすればよい。

また、指示を出す方法つまりリンクする方法は、テキストのような静的なデータへのリンクでなく動的なリンクが必要である。

7. おわりに

本報告では、動的に変化する情報の提供を含めたパワーディレクトリと分散サーバーによる情報の提供方法について述べた。また、マルチメディア情報提供の一例として試作したオンデマンドオーディオサーバーと応用例について述べた。

試作は特殊なハードウェアを使用せず、UNIX(Sun)とMacintoshというクロスプラットフォーム上にサーバー/クライアント形態で行った。性能については「クロスプラットフォームコンピュータネットワーク上での音声通信の実現」[4]を基づいて設計を行った。アクセストラフィックは6章で述べたように高音質であるほど高負荷となる。これは、高速LANでの利用を前提としたからである。そのため、LAN間接続等での低速回線での利用では品質が保てなくなる。このような場合にはQOS制御の必要が出てくるであろう。

今後は5章で述べたような応用技術への展開、6章での課題に取り組んでいく。また、分散サーバーによるパワーディレクトリの実現にも取り組んでいく予定である。

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 中村主幹研究員および関係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】

- [1] "The Internet Gopher software", The University of Minnesota, 1993.
- [2] Harry Morris, Brewster Kahle, Jonny Goldman, "WAIS", Thinking Machines Corp, 1992.
- [3] CERN, "World-Wide-Web", Switzerland, 1993
- [4] 篠田、石井、桑名, "クロスプラットフォームコンピュータネットワーク上での音声通信の実現", マルチメディア通信と分散処理研究会, 1993年7月.