

ATM LAN を用いたマルチメディア
インターネット ナビゲーション システム (MINS)

北村 浩、谷口 邦弘、坂本 浩充、板坂 大作*、西田 竹志

日本電気株式会社 C&C 研究所

日本電気株式会社 ULSI システム開発研究所*

kitamura,taniguti,sakamoto,d-itasak,nishida@nwk.cl.nec.co.jp

本稿は ATM LAN 上で動作する情報検索システムと動画再生機能を統合したシステムであるマルチメディアインターネットナビゲーションシステムの試作システムについて報告する。本システムはインターネット上のアプリケーションとして分散したサーバから送られてきた動画をリアルタイムに再生することを目的とし、WWW(World Wide Web)等の情報検索システムと組み合わせることにより、ユーザのビデオ情報検索を容易にしたシステムである。Audio と Video の同期を行なう動画処理系を持ち、ビデオサーバの分散的な配置にも対応している。動作環境としては複数のハードウェア、オペレーティングシステムで動作可能でマルチプラットフォーム環境に対応している。

A Multimedia Navigation System on ATM LAN :
Multimedia Internet Navigation System (MINS)

Hiroshi KITAMURA, Kunihiro TANIGUCHI,

Hiromitsu SAKAMOTO, Daisaku ITASAKA*, Takeshi NISHIDA

C&C Research Laboratories, NEC Corporation

ULSI System Development Laboratories, NEC Corporation*

This paper reports Multimedia Internet Navigation System (MINS) which integrates internet navigation system WWW and Multimedia on Demand over ATM LAN. The software components of the system consist of two subsystems. One is an information search subsystem and the other is a video transport subsystem. Especially the latter subsystem has three major functions, video control, inter-media synchronization, and transport control. The system can playout motion JPEG pictures with the size of 640x480 at the rate of about 25 frame per second, and achieves each media play-out synchronization and intermedia synchronization between audio and video.

1 はじめに

急速な CPU 性能の向上、ATM LAN をはじめとする高速なネットワーク環境の整備と共に、マルチメディアを用いた通信が盛んに用いられようとしている。また、インターネットの環境も急速に広がってきている。本論文では、これらの背景となる技術を統合してユーザの端末でマルチメディア情報を要求に応じて対話的に取り出せるシステム MINS (Multimedia Internet Navigation System) の試作を行なったので、報告を行なう。

マルチメディア情報の検索に関し、従来インターネットではナビゲーションの手段として WWW などを用いたシステムが知られている。しかし、現状の WWW の方式では問題がある。この方式ではサーバからビデオデータを取り出して見るには、一旦ユーザの端末にビデオデータを蓄積してから再生する方式をとっている。そのためには、ユーザの端末に大きな記憶装置が必要となる。更に、再生を始めるまでに長い時間待たされる。例えば 6Mbps に圧縮された 2 時間分のビデオデータは、5.4GB の容量になり、大きな記憶装置が必要で、転送するには 155Mbps の高速回線を利用しても約 5 分ほどかかり、その間ユーザは待たされてしまう問題がある。

そこで、この問題を解決する方法として今回開発した方式は、ビデオサーバとユーザ端末の間でデータを転送すると同時にユーザ端末上でビデオを再生する方式である。この方式では、ビデオデータの大きさによらず即座にユーザはビデオを再生でき、また、ユーザ端末側では大きな蓄積装置を用意する必要もない。

MINS は今後のネットワーク技術の発展を踏まえ、クライアント・サーバ方式のシステムとしている。本システムの開発目標を以下に示す [1]。

オープン性 特殊なプロトコルを使用することなく、インターネットの一つのアプリケーションとして動作する。

操作性 特別な知識のないユーザが操作できるような容易な UI (User Interface) を備える。

拡張性 将来他のシステムと統合ができるようにする。蓄積情報量が多くなったとしても分散的に情報管理ができるようにする。

システム構成を第 2 章で説明する。本システム MINS は情報検索系と動画処理系に大別されるが、それぞれ第 3 章と第 4 章で説明する。第 5 章では実装方式を、第 6 章では今回試作したシステムの評価について述べる。さら

に、第 7 章では今後検討していくべき問題について述べる。

2 システム構成

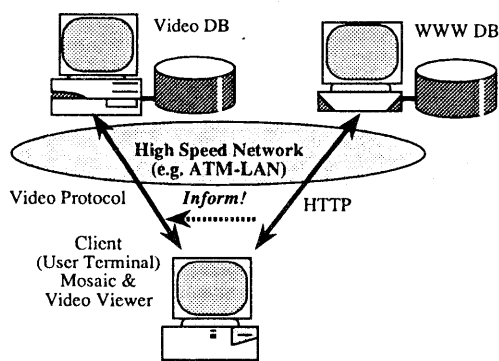


図 1: MINS のシステム構成

図 1 に MINS のシステム構成を示す [1]。本試作システム MINS は WWW をベースとした情報検索系と動画処理系からなる。これは、第 1 章で説明したオープン性と拡張性を満たし、情報検索系と動画処理系が独立にも動作するようにするためである。これにより、情報検索サーバと動画サーバは異なるマシンでも動作が可能であり、情報検索系サーバが異なるマシンの動画情報の参照を指示することもできる。情報検索系はユーザの要求するビデオ情報を検索することを支援する。動画処理系ではユーザの要求に応じてビデオサーバにアクセスし、所望のビデオデータを再生する。システムの論理構成は情報検索用データベースとそのデータベースを検索するプログラム、ビデオデータとビデオデータをユーザ端末に送信するプログラム、さらにユーザ端末では情報検索を支援するクライアントプログラムとビデオを再生するクライアントプログラムからなる。ユーザの情報検索の結果がビデオ情報である場合には、ビデオクライアントプログラムが自動的に起動され、ビデオを再生する。

図 2 に試作システム MINS のハードウェア構成を示す。端末として NEC 製 EWS4800、PC9821、SUN Micro Systems 社製 SparcStation 等を用いた。EWS4800 ではハードウェアによる動画像伸張ボードを使用しないので、無圧縮の画像情報を転送する。PC-9821 では MPEG のボードを用いる。SparcStation では Parallax 社製の Powervideo/Xvideo を用いる。サーバには NEC 製 PC-98/SV model 2 と SparcStation を用いる。ネットワーク Switch は NEC 製 ATOMIS 5 を用いた。各ホストのネットワークインタフェースは NEC 製の ATM

NIC (Network Interface Card) を利用している [2]。試作システムの外観を図3に示す。

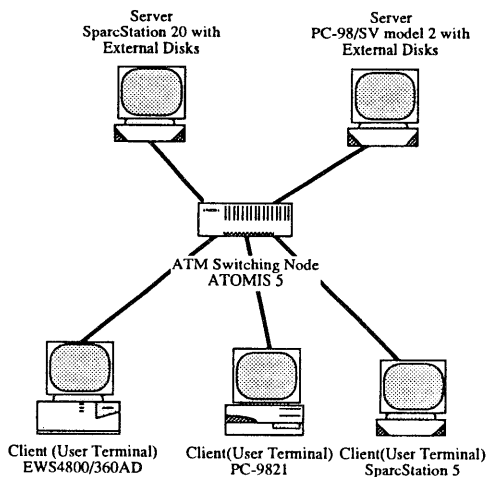


図 2: MINS のハードウェア構成



図 3: 試作システム MINS の外観

3 情報検索系

今回情報検索系としては WWW[3] を採用している。本システムでは WWW を拡張して新たなビデオサービスを行なうためのコンテンツ・タイプとして video/para を新たに定義している。このコンテンツはビデオサーバにアクセスするためのパラメータのみであり、ビデオサーバのアドレス、サービス番号、ファイル名、ビデオコーディング方式、オーディオコーディング方式、同期パラメータ、データ転送速度等の情報が含まれている。これらのデータは数十行のテキストデータであり転送処理の遅延はほとんど問題にならない。このパラメータを基に起動された external viewer はビデオサーバに接続し、クライアント側にデータをダウンロードしながら同時にビ

デオデータの再生を行なう。また、クライアントの機能の差を吸収するため、パラメータ情報はマルチクレーンフォーマットにしており、JPEG/MPEG/raw 等のデータが混在されたビデオサーバに対応できるようにしている。これらの拡張により、WWW とそのプログラムの枠組みを外れずにシステムを実装できるようになり、ソースコードの変更を行わず、コンフィギュレーションの変更だけで対応できる。また、パラメータ情報は CGI により動的に生成する。これにより、動画像データの管理等が柔軟に行えるようになる。また、サーバの過負荷を防ぐために同時にアクセスする人数を制限するロック機能、アクセス権認証機能も提供している。さらに、情報検索系のデータベースの核となる HTML では、クリックルマップを用いることにより美しい画像と使い易いユーザインタフェースを提供している。

4 動画処理系

4.1 ビデオプロトコルの概要

動画像系は前述の Mosaic から起動される external viewer とビデオサーバ側のプログラムから構成される。その間の通信は今回新たに開発したプロトコルに従って行なう。次に、ビデオプロトコルのサブスタックについて順に説明する。

ビデオ制御: クライアントとサーバの間でのセッション制御を行うプロトコルである。機能としては、コネクションの確立/切断、ビデオデータの選択、データ通信速度の指定、play、pause、quit、fast forward、rewind 等のビデオの再生モードの指定/完了通知、バッファのフラッシュ制御からなる。

同期制御: 再生時の音声と動画を同期させるための制御である。機能は、メディア内の同期とメディア間の同期、コマンドレベルの同期と同期情報の交換からなる。

転送制御: データをデータグラム化するためのデリミタに関する。特にバッファのフラッシュ制御や同期制御のためのトリガとして使用している。

次に通信形態の観点から説明すると、図4の様に本プロトコルは音声/動画/制御の3つの論理チャンネルからなる [1]。音声と動画のチャンネルはそれぞれ音声と動画のデータのみを転送する。音声と動画を混合して一つファイルとして蓄積され1本のストリームになるデータの場合はこれらのチャンネルを分離する必要はないが、本システムではフォーマットの一般性を個々のメディアが分

散して蓄積された場合のことも考慮に入れている。制御チャンネルには制御用のコマンドや同期情報などを転送する。クライアントからサーバに対して初めに開くチャンネルは制御チャンネルであるが、制御チャンネル上のコマンドにより音声と動画のチャンネルを開く。

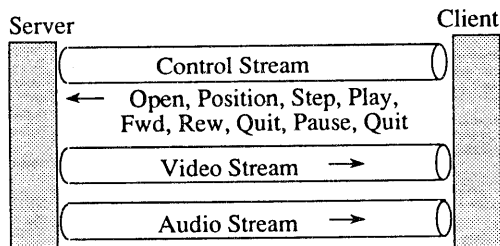


図 4: Video プロトコルの概略

4.2 ビデオ制御サブプロトコル

ビデオ制御サブプロトコルは、主にクライアントからのコマンドをサーバに転送することとそのコマンドの完了通知をクライアントに対して行うものである。コマンドには以下のものがある。

connect	セッションの開始
quit	セッションの終了
open	ビデオデータの指定
position	メディア間同期パラメタの指定
step	駒送り、若しくは、一フレームの送信
start	ビデオの再生開始
stop	ビデオの再生の停止
fastfwd	ビデオの早送り
fastbwd	ビデオの巻き戻し

これに対して以下の状態が存在する。

INIT	初期状態
Fullsynch	完全に同期をとるための状態
Pause	ユーザからのコマンド待ち状態
Play	再生状態
FWD	早送り状態
REW	早送り状態

サーバとクライアントは図 5 に示す状態遷移に従って動作する。ここで Full Synch は position コマンドによる 4.3.1 章で述べるコマンドレベルの同期機能を実現する状態である。position コマンドは任意のフレームの動画と任意の時刻の音声から再生するためのコマンドである。これにより動画と音声の同期を実現する。さらに動画像を途中から再生したり、飛ばしたりすることも可能である。step コマンドは Pause 状態内で使われるコマンドで

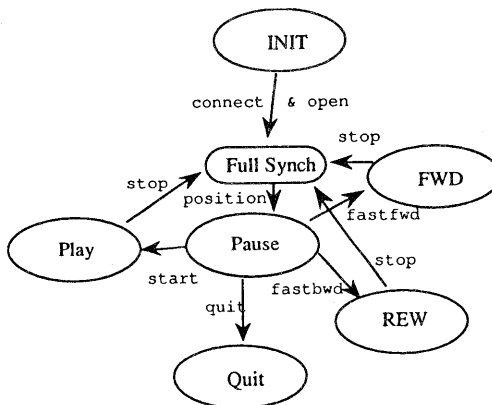


図 5: サーバとクライアントの状態遷移

ある。これは、external viewer の動作時に最初の 1 画面を表示するために使われている。start コマンドには引き数として動画像の駒落としの割合を記述できるようにしてある。これは将来的に輻輳が起きたときのアプリケーション側からのアクション等にも使うことができる。このようなコマンド体系を用いることにより VCR の機能をエミュレートすることができる。

4.3 音声と動画の同期

4.3.1 コマンドレベルの同期

図 6 に示すように、ある瞬間を見ると、クライアントがビデオの時刻 T の時点を再生中の時、サーバは時刻 $T + \alpha$ の時点データを送信することになる。このため何の制御もせずに、ユーザが rewind ボタンを押すとクライアントの画面にはサーバの送り出したデータの最後つまり時刻 $T + \alpha$ の時点まで画面を再生してから巻き戻しを始める。これはユーザにとって不要なデータを再生することになり、応答性も操作性も悪い。このため、コマンドレベルの同期が必要である。

コマンドレベルの同期は Pause 状態を利用して行う。この同期はクライアントからサーバに対して行われる。つまり、ユーザが何らかの操作を行った場合、クライアントはただちに stop コマンドを送る。さらに、再生中の動画の時刻をもとに position コマンドを送る。これにより、クライアントの再生中の動画時刻がサーバに通知され、サーバは動画と音声の時刻をそれに合わせ Pause 状態に遷移する。サーバは Pause 状態になったことをクライアントに伝える。クライアントでは受信バッファに溜まっていた全てのデータを廃棄し、ネットワーク中に残留するデータが存在しないことを確認して、クライアン

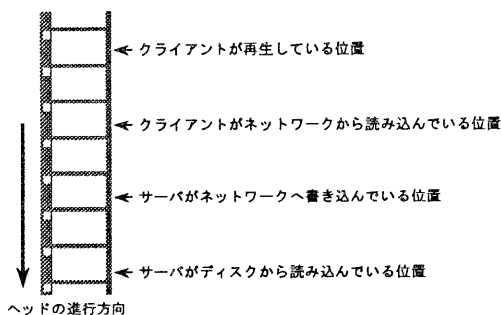


図 6: システムに存在する時間軸

とも Pause 状態になる。

4.3.2 ストリームとメディア間の同期方法の関係

メディア間の同期とは複数のメディアの間の時間軸の関連づけを行うことである。考えられる手法としては以下のものがあるが、これらはチャンネル上の通信方式と深く関わる。

1. Audio と Video のデータを交互に混合し一つの連続するストリームに流し、再生するときは交互に再生することにより同期をとる。この方法の特長は伝送路上でも完全な同期が得られることである。しかも、実装も非常に容易である。欠点としては、音声と動画の情報が同じ通信路を通るために、基本的に同じ通信品質での通信しかできない。一般的には Audio と Video に要求される通信品質は異なる（例えば、ネットワークが輻輳した状態において、Video データには欠落が許されるが、Audio データには欠落を許さない場合がある）ので、今回はこの方法を利用しなかった。将来パケット毎に通信品質を変更することができるようになればこの方法は有力な方法となるであろう。

2. RTP(Real-time Transport Protocol)[5] の様に全てのデータグラムパケットにタイムスタンプを持ち、その時刻を利用しながら再生を行う。この方法の特長は全てのパケットに再生時刻が記されているため、時刻同期が容易である。さらにネットワークの遅延の評価等も同時に行え、それにより通信パラメータを再調整することもできる。また、音声と画像を異なるチャンネルにより送ることができる。（必要であれば画像の低周波成分と高周波成分を別々のチャンネルで送ることもできる。）しかし、これを実

現するプロトコルは技術的な開発段階にあり、現時点ですぐに用いることができない。

3. Audio と Video データは異なるストリームで転送する。Audio データは CBR (Constant Bit Rate) であるため再生済みのデータ量から時間情報を得ることができる。Video データの流れは VBR (Variable Bit Rate) であるが、単位時間に含まれる frame 数は一定である。そのため再生した frame 数を数えれば時間情報を得ることができる。得られた両方のメディア間の時刻情報を基にメディア間の同期を実現する。この方法の特長は Audio と Video を異なるストリームで転送するため各メディア毎に異なる転送転送プロトコルを用いることで、QoS の異なるデータ転送をすることが可能である。例えば下位層に TCP を用いれば信頼性とフロー制御をポート番号による多重化によって容易に実現できる。以上の理由から本試作ではこの方法を用いている。

4.4 転送制御サブプロトコル

今回は通信の信頼性を容易に確保するためにデータの転送時にも TCP コネクションを用いている。TCP コネクションはバイトストリームの信頼性を保証するプロトコルである。これに対して、バッファのフラッシュ制御や同期制御のため上位サブレイアが要求するチャンネルタイプはデータグラム型である。それゆえ、TCP をデータグラムプロトコルに見せるためのレイアが必要となる。本システムでは図 7 の様にチャンネル上のデータフォーマットをデータとそのデータの長さの組という形にしている。さらに、コマンドレベルの同期をとるためのデリミタを付けている。バッファのフラッシュが必要な場合はコンピュータ上のバッファを空にするだけでなく、このデリミタが読めるまでネットワークからデータを読み、廃棄した。

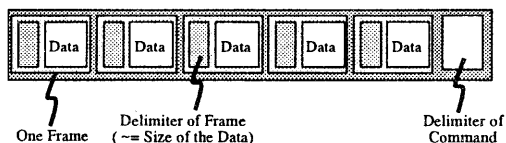


図 7: ストリーム上のデータデリミタ

5 実装方式

5.1 高速化方式

動画像転送/再生を行う場合のボトルネックとして考えられるものは以下の3つである。

1. 蓄積動画像データをディスクから読み込む処理
2. 動画像データをネットワークで転送する処理
3. 受信した動画像データを再生する処理

1は高速で安定して読めるディスクのシステムを用いて解決する。

2は高スループットのネットワークを用いることで解決出来る。今回は ATM LAN を利用することで解決する。また、ATM LAN の性能を引き出し、プロトコルによるボトルネックが生じないよう、ウィンドウサイズなどの TCP/IP のチューニングも行なう。

3は動画の扱うソフトウェアやハードウェアに依存するが、画像再生を高速に行なうためには、クライアントの構造をネットワークの入力と動画像の再生処理でそれぞれブロックしてしまうことにより、互いの処理を妨げることがないように構造にすることにより解決する。これを実現する方法としてそれぞれのクライアントのハードウェア構成に応じて以下の二つを採用している。

1. 画像再生処理がブロックされないようにする。そのため、フレームバッファに直接データを書き込むようにしている。
2. 画像再生側のハードウェアが再生処理をブロックする関数で行なっている場合は個々の処理をマルチスレッドで実現する。

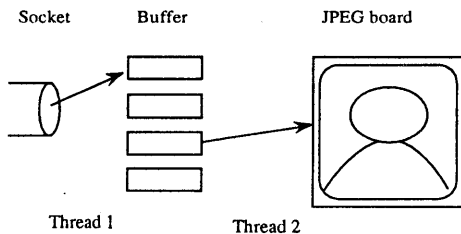


図 8: multi thread/multi buffer の描画

5.2 ファイルフォーマット

ファイルのフォーマットは動画像の Fast Forward、Fast Backward に対応するため、ファイルを順/逆方

向から読んでもフレームの切れ目がわかるようにしている。特に JPEG 圧縮/伸張を利用する場合は、1 フレームあたりのバイト数が可変である。それゆえ、フレームデータのサイズ (4 バイトの整数)、フレームデータ、フレームデータのサイズの三つをまとめて 1 フレームデータとした。この 1 フレームデータを連ねることにより、動画像のデータとしている。音声データはヘッダにコーディング方式、サンプリング周波数、量子化ビット数、チャンネル数の情報が記述する format を用いている。

6 システムの評価

まず、ユーザインタフェースであるが、これは Mosaic をそのまま利用しているので、マウスのクリックだけで全てコントロールできる。我々が開発したビデオサービスのための external viewer が他の情報と同様にマウスのクリックだけで表示されることは、情報検索系と動画処理系とのインタフェースが一般化された形態をとっており、本システムの開発目標であるオープン性を満たすものである。一秒間に再生されるフレーム数と再生されるフレームと音声との間の遅延について述べる。

表 1: MINS のクライアントでの総合評価結果

	無圧縮画像	Motion JPEG
画像サイズ	360x242 (表示は縦横 2 倍)	600x400
size/1frame	約 90KB	約 22KB
frame/sec	20fps	約 25fps
bit rate (1.5Mbps Audio 含む)	約 16Mbps	約 6Mbps

無圧縮画像の場合、360x242 程度の画像をクライアント側で縦横 2 倍の大きさに拡大しながら再生するが、現実的に再生可能な速度は 20fps である。この原因はサーバにおけるハードディスクからのデータの読み込み速度に起因する。実測ではハードディスクのアクセス速度は約 2MB/s である。これ以上速度を上げると短時間だけ正常に動作するが、実行的ディスク読み込み速度が低下し、Audio と Video の同期がずれることがある。なお、クライアント側について述べると理想的にネットワークからデータが入ってくる時は 50fps 以上の描画速度が確認されている。

Motion JPEG の場合、約 25fps での再生を行えた。この場合のボトルネックは JPEG decode の速度である。

マルチスレッドを利用した効果はJPEG表示の関数が実質的にブロックされる関数であることや、スレッドスイッチングがselect関数の処理よりもはるかに速いこと、readに費やすCPU時間からわずかに速くなっていることより、示すことができる。マルチスレッド化による利点は性能だけでなくプログラミングの容易さとそれによる開発工数の短縮にある。動画と音声の同期に関しては、主観評価であるが、同期が取れてなくて不自然に見えるという反応はなかった。サーバの性能は、前述のとおり、ハードディスクの読み取り性能に依存しているが、本システムは複数台のサーバを許す構造なので、サーバ間の負荷分散も可能である。

7 課題

今回の試作により様々な問題が解決された。しかし、残された課題もあり以下に列挙する。

7.1 トランスポートプロトコル

今回トランスポートプロトコルとしてTCPを利用している。現状では適した方法であるが、将来にわたって最適な解とはいえない。

MINSの動画処理系のようなマルチメディア通信のアプリケーションではバッファのフラッシュ処理や画像表示処理のためにはデータグラム型が適しているの、MINSではデータはデータグラム型で扱っている。そのためデータグラム型のプロトコルが適していると考えられるが、現状ではデータグラム型で、信頼性を保証するプロトコルは存在しない。また、時間の情報を伝える標準的なプロトコルも存在しない。そこで、今回は過渡的な手段として信頼性を保証するプロトコルであるTCPの上にデータグラムを載せる方法を用いており、必ずしも最適ではない。今後は、現在研究開発段階にある必要な機能を有したマルチメディア通信に対応するプロトコルを実装していくことが必要である。

7.2 データの性質と同期の方式

今回は蓄積型のデータの同期を実現している。蓄積型のデータは遅延の絶対量を小さくする必要はなく、遅延のジッタを最小化すればよい。これに対して、会話型アプリケーション型データのメディアの同期には、遅延の絶対量も最小化しなければならないという制約項が加わる。この型のデータの同期を実現するには新たな同期方式が必要である。

7.3 マルチキャストへの対応

複数のユーザが同時に同じメディアを再生するという事も考えられる。特にグループウェアでの枠組み内でビデオサービスを利用することも考えられる。マルチ

キャストを行った場合の同期制御や効率的なトランスポートプロトコルも今後の研究課題である。

7.4 品質管理

動画サーバの能力、ネットワークの込み具合に応じた動画品質の制御機能を組み込むべきである。本システムではクライアントが受信データ速度を決定し、サーバへ通知している。しかし、再生品質はサーバとクライアントとネットワークの負荷状況を見て、必要であれば、CPU使用時間やネットワークの帯域を予約して、サービスの品質を動的に決定する必要がある。

8 まとめ

ATM LAN上で動作する情報検索システムと動画再生機能を統合したシステムであるマルチメディアインターネットナビゲーションシステムの研究開発に関する報告した。また、本システムのようなマルチメディア情報をリアルタイムに交換する際に解決しなくてはならない課題について報告した。

今後トランスポートのプロトコルとしてマルチメディアに向けたプロトコル、例えばRTP[4][5]などの実装を行なう予定である。さらにはマルチメディア通信するには不可欠な高速通信機構である0-copy architecture[6]をこのシステムに適應させることなども予定している。

参考文献

- [1] 谷口他 “インターネットナビゲーションを用いたMMoDシステム”、情報処理学会 95年春季全国大会
- [2] 坂本他 “ATMホスト・インタフェース・ドライバの開発”、電子情報通信学会 94年春季全国大会
- [3] R. J. Vetter “Mosaic and the World-Wide Web”, IEEE Computer, October 1994.
- [4] Sculzrinne, Casner, Frederick, Jacobson “RTP: A Transport Protocol for Real-Time Applications,” draft-ietf-avt-rtp-07
- [5] 北村他 “マルチメディア通信プロトコルRTPの機能拡張と試験的実装”、電子情報通信学会 95年春季総合大会
- [6] 北村他 “UNIXワークステーションにおける高速プロトコル処理実装方式”、電子情報通信学会 交換システム研究会 93.11