グループ共同作業におけるマルチメディアナビゲーションシステム-MINS

立川仁也 谷口邦弘 北村浩 西田竹志

日本電気株式会社 C&C研究所

本稿はマルチクライアント，特に会議システムでの情報検索システムと動画像再生機能を統合したマルチメディアナビゲーションシステムMINSの構成と，そこにおけるマルチメディアプロトコル，メディア同期方式について報告する．本システムは，サーバに蓄積された任意のマルチメディア情報をそれぞれのユーザ端末間でメディア同期し，高品質に再生MMoD(Multimedia on Demand)を実現する．本システムでは，複数クライアント間で動画再生制御の同期をとるための状態管理方式，ネットワークに特性による動画／再生品質の劣化を最小限にするバッファリング方式，およびタイムスタンプ制御を用いたメディア内／間同期手法を用いている．本稿では上記方式およびリアルタイムプロトコルRTPの実装方式について述べる．また，実システム上でのパケット遅延・損失特性と再生品質との関係について評価する．

1　はじめに

通信の高速／広帯域化，またMPEGのような高品質な動画圧縮方式の標準化により，MMoDの利用環境が整い始めてきている．またインターネットは世界的な広がりを見せ，コンピュータネットのインフラとして整備されてきている．筆者らはこのような状況を鑑み，インターネットナビゲーションとMMoDを統合したシステムMINS (Multimedia Internet Navigation System)を開発し，報告してきた[1, 2]．本システムは高速なATM網を利用し，情報検索と動画再生機能を持ち，ユーザのビデオ情報検索を可能にしたシステムである．ネットワークに蓄積してある動画情報をリアルタイムに動画再生を行なえることを目標としている．サーバーからビデオデータを取り出して見るには，現状のWWW(World-Wide Web)のような方式では，一旦ユーザーの端末にビデオデータを蓄積してから再生するため，端末側で大きな記憶装置が必要になり，かつ再生を開始するまでに長い時間待たされてしまうという問題がある．本システムではサーバ／クライアント間のプロトコルでフロー制御することにより，クライアントで受

Multimedia Internet Navigation System for Group Cooperative Work - MINS

Hitoya TACHIKAWA, Kunihiro TANIGUCHI, Hiroshi KITAMURA, Takeshi NISHIDA

C&C Research Laboratories, NEC Corporation

信と同時に再生できるようにしている．

本システムでは，さらにグループウェアと統合することによって，テキストやイメージだけでなく動画情報も共有し，共同作業における円滑なコミニュケーションを実現できるようにしている [3]．

グループ共同作業においては，複数のユーザが，同時に同じ動画情報を共有し異なった再生制御を行なう可能性がある．このため複数ユーザ間での排他制御および同じ画面を同時に見ることができる再生同期制御が必要になる．また音声と画像が，異なった回線で送られてきた際にそれらの間の同期制御 (メディア同期) が必要になる．さらにネットワークでのパケット遅延の jitter が生じるため，再生品質を保つためのバッファ，クライアント／サーバ間のフロー制御が不可欠である．

本稿では上記システムをベースとして，会議等の共同作業において，複数のユーザ間で動画情報の共有方式，および本システムにおけるマルチメディア通信プロトコルとしてインターネット準標準の RTP (Real-time Transport Protocol) 用いたマルチメディア同期再生メカニズムを中心に報告する．

第 2章では，情報検索系と動画再生系で構成される本システムの概要を説明する．第 3章はプロトコルスタックについて述べる．第 4章はメディア同期制御について述べる．第 5章では本システムの評価を述べる．

2　システム構成

本システムは (図 1) に示すようにマルチメディア情報を蓄積するサーバ，それにインタラクティブにアクセスする複数のクライアントから構成される．またソフトウェア構成としては情報検索系と動画処理系からなる．情報検索系はユーザーの要求するビデオ情報を検索することを支援する．動画処理系はサーバ側での動画ファイル管理とネットワーク制御処理とクライアント側のネットワーク制御・再生処理より構成される．

クライアント側にはユーザインターフェイス機能があり，参加者は番組選択画面を用いて所望のビデオ情報を選択し，サーバからの情報は動画再生画面に表示する．またコントロールパネルを用いて再生状態の変更を通知する．

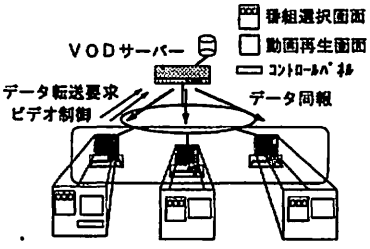


図 1: システム概略

2.1　情報検索系

クライアントの情報検索系はインターネットナビゲーション Mosaic，サーバの情報検索では WWW を用いている．このため既存のインターネットナビゲーションシステムとリンクして利用可能である．さらに，MINS サーバにはビデオサーバにアクセスするためのパラメータを格納している whois サーバーを備えている．この whois サーバーはビデオサーバーのアドレス，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書かれている．サーバ／クライアント間の操作は以下のようになる．

1. MINS クライアントは選択された動画を一意に特定する nickname(例えば題名) を用いて whois サーバーに問い合わせる．
2. whois サーバーがその動画に関するパラメーターを返す．
3. このパラメーターを基に起動された external viewer はビデオサーバーに接続する．
4. 接続されたビデオサーバーは whois サーバーに問い合わせ，送信に必要なパラメーターを得て，データの送信をクライアントに行なう．

5. データを受けとった external viewer は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.

この whois サーバーのパラメーター情報は，データが混在されたビデオサーバに対応できるように，マルチフレームフォーマットにしている.

2.2 動画処理系

動画処理系は，クライアントのビデオ再生表示を行なう External Viewer とビデオ再生制御処理と，ビデオサーバーのサービス処理から構成される.

External Viewer はビデオサーバーから送られてくるビデオデータを再生する．ビデオ再生制御は VCR 並の対話処理を提供するもので，ユーザーからの操作に応じた再生制御コマンドの送信を行なう．対応するユーザインターフェースとしては play, pause, quit, fast forward, rewind 等再生の状態の変更を行なうボタンや，任意のフレームに jump するスクロールバーなどがある．図 2に画面例を示す.

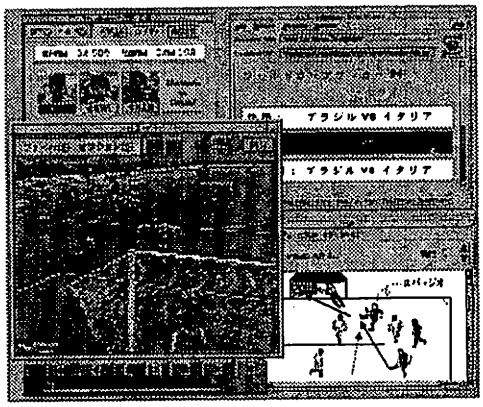


図 2: MINS 画面例

3 プロトコルスタック

MINS におけるプロトコル構成を図 3に示す．ビデオ制御プロトコルは，クライアントとサーバー

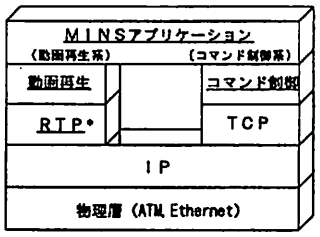


図 3: プロトコルスタック

の間でセッションの制御を行な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，からなる.

次に通信形態の観点から説明すると，図 4のように本プロトコルは音声／動画／制御の 3 つのストリームを開設し利用する．音声と動画のチャ

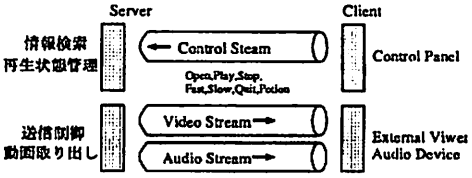


図 4: ストリーム構成

ネルはそれぞれ音声と動画のデータのみを転送する．クライアントからサーバーに対して初めに開くチャネルは制御ストリームであるが，制御ストリーム上のコマンドにより音声と動画のチャンネルを開く.

3.1 ビデオ制御プロトコル

3.1.1 状態遷移

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ーに転送すること．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
play	通常再生
fast	早送り再生 (逆もあり)
slow	スロー再生 (逆もあり)
position	現在表示の frame 番号表示

これに対して以下の状態が存在する.

INIT	初期状態
PLAY	再生状態
STOP	再生停止状態
FAST	高速再生状態 (逆もあり)
SLOW	低速再生状態 (逆もあり)

サーバー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. クライアントが起動され, サーバーに接続 (connect) し, video と audio の Stream が張られる (open). そして, 再生が開始 (play) され, ユーザーの選択通りに停止 (stop), 早送り (fast), スロー再生 (slow) を行なう. 早送りとスロー再生では, 引数に speed-factor があり, 正／負の整数で順送り／巻きもどし制御を指定する. また現在再生している frame 番号を問い合せ (position) を行なう. quit ボタンが押されることで終了 (quit) になる.

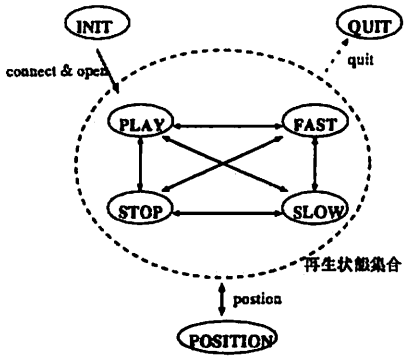


図 5: 状態遷移

3.1.2 再生状態遷移の一元管理

グループ共同作業における動画情報の再生では, サーバ／クライアント間は 1 対多の関係であり, 参加者間で動画像の共有を行なっている. このため, 各クライアントでの再生制御状態は同じでなければならない.

そのため再生状態をサーバーが一元管理するようにしている. さらに, サーバからは再生状態に応じた転送速度でデータを送る方法にしており, 各クライアントは再生がどの状態にあるかを知る必要がなく, 送られてきたデータをそのまま再生するだけで良い.

3.2 トランスポートプロトコル

マルチメディア情報のリアルタイム転送／再生という観点からトランスポートプロトコルを考えると次の 2 点を注意する必要がある.

1. メディア同期の容易性
2. フロー制御

また複数クライアントとするとマルチキャスト性を考える必要がある. 1 の点では MPEG トランスポートは最適であるが, インターネット環境ではそのまま転送できない. なぜならば, コストが高い・音声／動画を多重化して同じ DB に蓄積する必要がある, という制約があるからである. このため, マルチメディア符号化に依存せずインターネット環境でも通信できる一般的な転送プロトコルを用いる必要がある.

ここで考えられるのは, インターネット標準の TCP/IP スタックであるが, 既存のトランスポートプロトコル (例えば TCP, UDP) は, マルチメディア通信を行なうには機能が足らない. TCP ではサーバ／クライアント間にコネクションを張り, データ転送に, ウィンドウベースフロー制御／再送制御管理を行なっている. しかし, マルチキャストを行なうには複数の相手とのコネクショ

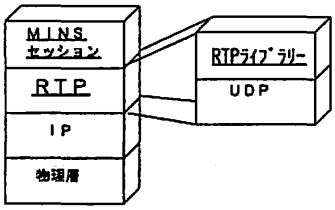
ンを管理しなければならず，制御が複雑となる．また，UDP はマルチキャストは容易であるが，フロー制御がなく信頼性が低い．

そこで今回，現在インターネットで標準化準備中の RTP(Real-time Transport Protocol) を用いメディア同期のメカニズムをアプリケーションで実現している [4]．RTP の特長としては

1. ヘッダー内に timestamp,sequence number のフィールドが設けられ，受信側で再生時刻制御，フレーム分割転送が可能．
2. メディア符号化がサーバ／クライアントで調停可能．
3. マルチキャストが可能．

がある．このため複数のメディア間の時間軸との関連づけが行なえ，データの同報も容易に行なえる．

本システムでの実装形態は，UDP 層の上にライブラリーとして実現している．また UDP/IP のマルチキャスト機能を利用して，複数参加者へのデータの同報が行うことができる (図 6)．



下線部分ー今回実装した所

図 6: ネットワーク階層

さらに，本システムでは RTP に対してフロー制御として，レート制御が可能であるような機能拡張をしており信頼性の高い通信が可能となっている．

4 メディア同期方式

高品質なマルチメディア再生では，関連したメディア間の時刻同期を図ることが重要である．本システムでは，RTP での時刻情報とネットワーク遅延／jitter 吸収のためバッファ制御を元に，以下のようなメディア間同期方法を実現している．

サーバでは，音声の RTP パケットとそれに対応する動画の RTP パケットのそれぞれヘッダーの timestamp フィールドに，サーバの時刻情報から生成した値を入れて転送する．

クライアントでは，音声・動画再生はクライアントの時刻情報と RTP パケットのヘッダーに付いている timestamp を用いる (図 7)．ネットワーク

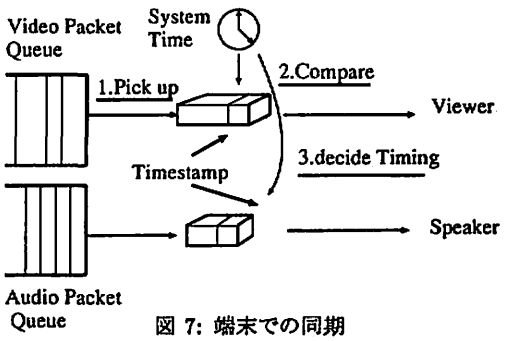


図 7: 端末での同期

遅延の分散を吸収するために，再生開始時刻は最初に音声パケットを受けとった時間から，わずかな時間分遅らせておく．音声・動画のパケットそれぞれは，別々の Queue に蓄積されている．

遅延差吸収をするために，以降の音声・動画パケットの再生は，Queue からパケットを取りだし，その timestamp の値を見て，再生すべき時刻になったら再生デバイスに書き込む．さらに，その値から，次に Queue から取り出すべきデータの時刻を計算しておく．

Queue から検索した際に期待されるデータがない場合，音声では直前のデータを再度デバイスに送る．また，動画は何も処理をしない (フレームバッファ内の動画データが表示されたままになる)．それぞれの Queue からパケットを取りだし，その timestamp 値を見て，再生すべき時刻と比較し，フレームを再生デバイスに書くタイミングを決定する．もし，遅すぎるならばそのフレームを廃棄

し，前フレームを再度表示する．早過ぎるならば再生すべき時刻まで待たせてデバイスに書き込む．

この方法により，ネットワークのjitterやパケットの欠落をそのままデバイスに伝えず，一種のフリーズ(freeze)機能を利用することで，なるべく再生品質を落さず音声と動画の同期をとりながら再生を行なえるようにしている．

また，各端末間であらかじめ実時刻の同期をとっておけば端末間の同期も正確にとることができる．

5 システムの評価

5.1 全体評価

まず，評価に使用したハードウェア構成を示す．サーバーには，SUN Microsystems社製 SparcStation Model 20，端末には，SparcStation Model 5を用いている．端末の動画処理ボードでは，Parallax社製のPowervideo/Xvideoを用いた．またネットワークはNEC製ATOMIS 5を用いたATMネットワークを使用している．各機器のネットワークインターフェースは試作したATM NIC (Network Interface Card 物理速度155Mbps)を利用している．

ユーザーインターフェイスであるが，情報選択画面にMosaicを，コントロールパネルにX WindowのGUIを利用しているので，マウスのクリックだけで全てコントロールできる．これは情報検索系と動画処理系とインターフェイスがユーザーにとって統一されているされているために，使いやすいものとなっている．

次に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．今回は素材としてMotion-JPEGのみの再生を行ない，音声／動画通信プロトコルとしてRTPとTCPを使ったケースとの比較評価も行なっている．

最初に十分帯域がある状態で，TCPでは約25fpsで再生を行なえた．一方，RTPでも同様に約25fpsで再生を行なえた．ここでのボトルネックはJPEG decodeの速度である．

なお動画と音声の同期に関して主観的な判断であるが，両方ともに不自然に見えるという反応はなかった．

5.2 再生品質に対するjitterの影響

5.2.1 実験方法

ネットワークから送られてくるデータパケットには，ネットワーク遅延／jitterが存在する．TCP/RTP/ UDPそれぞれのjitterに対する特性を，本システムにて比較評価するために以下の実験を行なった．

まず，人為的にネットワーク遅延／jitterを発生させるために，動画の転送レートぎりぎりにATMコネクションの帯域を制限し，サーバーから同じクライアントに別のプロセスにて無関係のパケット送信を行なう．これにより動画転送がじゃまされ，転送時間にずれが生じる．送信を行なうトランスポート層として，RTP・TCP・UDPの3つを用い，それぞれ，上記のような負荷をかける状態とそうでない状態での遅延／jitterの様子を計測した．負荷は，1分04秒の長さを持つ動画再生中に，再生開始から20秒経過してから10秒間サーバーからクライアントに送る．

5.2.2 RTPとTCPの比較

図8は，TCPとRTPの時の結果の図である．横軸にはTimestampの値，縦軸にはパケット到着期待時間と実際に到着した時間との差をとっている．図中Aで負荷をかけ始め，Bにおいて負荷を終了している．図8から，負荷状態では，RTPのほうがTCPよりもjitterが小さいように見える．

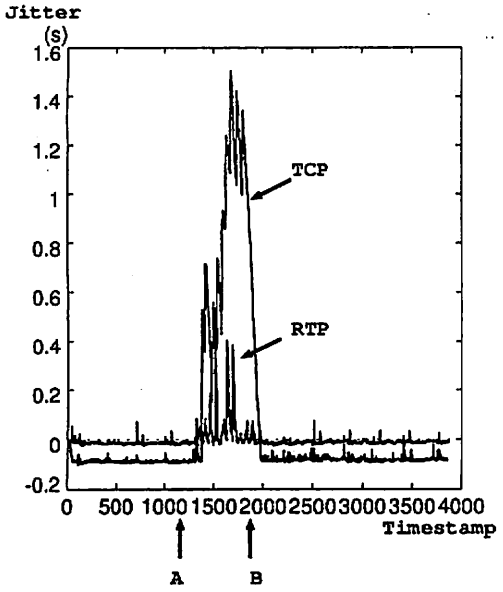


図 8: TCP と RTP の jitter

しかし，RTP ではパケット破棄が起こっているために，届かないパケットがありそれが観測されなかったためにこのように見えている．

TCP では，無負荷状態では，動画再生状態は良好であった．負荷がかかり始めると再生画面がフリーズ状態になり，負荷をかけなくなるまでそのままであった．これに対し，RTP では，無負荷状態では，動画再生状態は TCP と同様に良好であった．負荷がかかり始めると frame が落ちるのがはっきりわかるが，負荷中でも一部のフレームは再生できていた．

これは TCP の場合，負荷がかかったら，データの再送が発生してくる．再送パケットのためサーバ側では本来の時刻に送信すべきパケットがじゃまされてしまう．また TCP の再送制御は Go-back-N であるため，この傾向はさらに増幅される．このためクライアントでは，Queue に frame データに入れられるが，再生すべき時刻を過ぎているためにその frame データは同期制御で捨てられてしまう．そのため viewer はフリーズ状態のままになってしまったと考えられる．

このことから，一旦ネットワークに輻輳が発生しはじめると，TCP では再送機能が働いてデータは確実に届くけれども，そのデータの再生すべき時刻を過ぎているために，そのデータには価値がなくなってしまうことが言える．TCP の再送機能は，リアルタイムにデータを再生するには過剰な機能と言える．

5.2.3 RTP と UDP の比較

さらに RTP の廃棄の影響を調べるために，次に RTP と UDP を比較した．RTP では，再生状態は上記で述べたようになったのに対し，UDP では，無負荷状態では，動画再生状態では TCP, RTP と同様に良好であった．負荷がかかり始めると再生は行なえていたが，画面がフリーズする長さ，頻度が RTP の時より多くなった．図 9はその負荷状態の時間帯を拡大して示している． time-

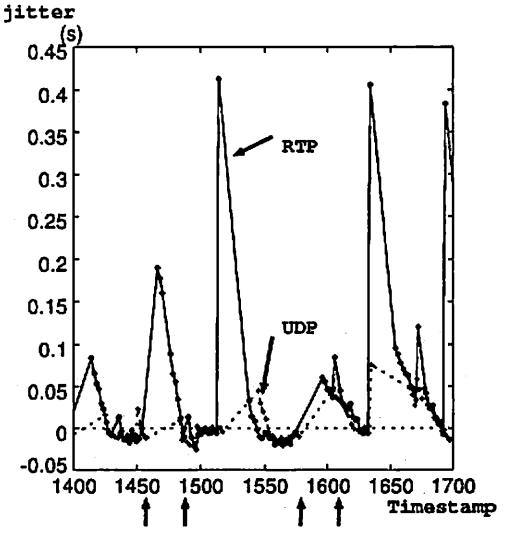


図 9: 負荷状態での UDP と RTP の jitter

stamp の 1459 ～ 1481,1579 ～ 1605 の部分が，UDP では画面がフリーズした箇所である．UDP ではパケット欠落が集中して起こっているため，この部分は jitter が生じていないように見えている．一方 RTP は 0.4 秒の jitter でパケットが届いている．

すなわち，UDP では，この時データが送られてこないため再生を行なえず再生画面はフリーズ状態であった．対して，RTP ではこの jitter を吸収して再生を行なえたことが言える．

UDP でパケット欠落が起こるのは，パケットが集中してサーバから送り出されているため，ネットワーク途中の buffer などでパケットの溢れが起き，パケットが集中的に破棄されていると考えられる．一方，RTP ではレートベースのフロー制御を行なっているために，パケット溢れが起きにくく，パケットの集中破棄は起きにくくなっていると考えられる．

以上の結果より，マルチメディア情報をリアルタイムに再生する場合に不必要なデータの再送は行なわず，レートベースのフロー制御を行なう方法が優れていると考えられる．

6 まとめ

今回開発したシステムはネットワーク上でのグループウェアプラットフォームである当社開発の ATM-MERMAID の一機能としても動作している [5]．本システムは「グループウェアにおける動画情報の共有」という新しいサービスの可能性を示した．

本稿では ATM LAN 上で動作する情報検索システムと動画像再生機能を統合したシステムであるマルチメディアナビゲーションシステムの開発に関する報告をした．その中で特に複数のクライアントで同じ動画像を共有するための再生制御方式，およびメディア間同期をとる高品質なメディア再生を行なうためのトランスポートプロトコル RTP の実装／評価について述べた．今後は，例えば 0-copy architecture[6] を本システムに適応させることで，本システムを用いて，より品質の高く高機能なネットワークプラットフォームの研究を行なっていく予定である．

参考文献

[1] 谷口邦弘, 北村浩, 坂本浩充, 板坂大作, 西田竹志:「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] 北村浩, 谷口邦弘, 坂本浩充, 板坂大作, 西田竹志:「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)」情報処理学会 第 21 回 情報メディア研究会,1995

[3] 水野浩三, 福岡秀幸, 前野和俊:「マルチメディアオンデマンドとグループウェアの統合」情報処理学会 95 年春季全国大会

[4] Sculzrinne, Casner,Frederick, Jacobson: "RTP:A Transport Protocol for Real-Time Applications" draft-ietf-avt-rtp-07.ps,1995

[5] 水野浩三, 福岡秀幸, 谷口邦弘, 立川仁也, 坂上秀和, 川崎成人:「グループウェアとマルチメディアオンデマンドを統合したマルチメディアオフィスシステム」電子情報通信学会 オフィスシステム・画像工学研究会,1995/9

[6] Kitamura, Taniguchi, Sakamoto, Nishida: "A New OS Architecture for High Performance Commnication over ATM Networks - Zero-copy Architecture" NOSSDAV'95