

静岡大学 博士論文

モバイルコンピューティング環境における
マルチメディア通信方式に関する研究

平成 10 年 6 月

大学院理工学研究科

設計科学専攻

太田 賢

論文要旨

携帯電話、PHSなどの広域移動無線通信インフラの発展と、携帯端末の高機能化を背景に、移動中や出先からでも必要時にインターネット、イントラネットにアクセスし、情報発信、受信を行うモバイルコンピューティングが注目を集めている。モバイルコンピューティング環境においてマルチメディア情報を扱いたいという要求が、ビジネス、エンターテインメント、公共サービス、それぞれの分野で存在する。

ビデオ、オーディオのような実時間制約を持つ連続データの通信を行う場合、通信路は連続データが要求する通信サービス品質（データ転送速度、誤り率、遅延など）を保証しなければならない。しかし、無線通信には低速、不安定、スループットや遅延の変動といった通信サービス品質の課題がある。インターネットもベストエフォート（最善努力）という概念に基づき、基本的に品質保証の仕組みを持たないので、輻輳や負荷の増大により通信サービス品質は低下する。

さらに、モバイルコンピューティング特有の問題として、携帯端末の移動性、切断性という課題を考慮しなくてはならない。セルラシステムはハンドオーバー技術により端末がセルをまたがって移動することを可能にしているが、それに加えてインターネット上の移動もサポートする必要がある。また、ユーザの都合、通信路の状態、回線接続の場合の通信コスト、端末のバッテリーの制限などの様々な要因から、携帯端末はネットワークから切断する機会が多く、切断を前提としたシステム設計を行う必要がある。

本論文は、通信サービス品質の課題とモバイルコンピューティング特有の課題を解決するマルチメディア通信方式として、同期・非同期通信方式、選択的マルチメディア通信方式と先読み方式、実時間指向優先転送方式を提案し、その実装、評価をするものである。

同期・非同期通信方式は、ビデオ会議システムやビデオ電話などにおけるヒューマンコミュニケーション(会議と呼ぶ)が、ユーザの都合や通信の切断により中断されるという問題を改善する。会議形態を同期、非同期に切り替えることのできる自由度、会議情報の蓄積によってユーザが同期会議から一時的に欠席することを許す柔軟性、ユーザの代理受信を行うサーバの導入によって切断中のユーザへ送信することを可能にする利便性、を提供する。会議実験により会議情報の蓄積が会議の欠席者に対し有効であることが分かった。

選択的マルチメディア通信方式は、蓄積型のビデオコンテンツの作成者、提供者が、意味的重要度に基づいてコンテンツの各シーンに優先度を与え、その優先度に従ってユーザに重要シーンを優先的に転送するものである。実験により、低速通信環境では低優先度の情報を落とすかわりに、重要な情報を優先的にユーザに提供できることが分かった。先読み方式は、蓄積型ビデオの取得の際、受信者に常に時間的に先のビデオ情報を与え、ストレージに保持させる。一時的に転送が途切れても、ビデオ再生を続行できることが分かった。

実時間指向優先転送方式は、経路上の各ノードでパケットの優先度と実時間制約に基づいたパケットスケジューリングを行うものである。シミュレーションにより、低負荷時には実時間通信を達成できること、高負荷時には低優先度のパケットの実時間通信は達成できないかわりに、高優先度のパケットの実時間制約が守られることを確かめた。低速通信環境でも効率的な転送を行うことができる。

これらの通信方式の適用例としてモバイル電子会議システム考え、同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式がモバイル電子会議システムの構築に役立つことを示した。また、各通信方式を組み合わせて利用した際の効果、問題点についての考察も行った。

最後に、マルチメディア通信システムを開発するためのツールとして、実時間プロトコル設計言語とそのシミュレータについて紹介する。実時間プロトコル設計言語を用いて、実際にモバイル電子会議システムの会議形態の切り替えのロジックの記述、実時間指向優先転送方式の設計仕様を記述し、シミュレータにより動作確認、性能評価を行った。その結果、実時間プロトコル設計言語とシミュレータの利用は、実装を簡単にし、設計の早い段階でマルチメディア通信システムの問題点を明らかにできることが分かった。

目次

1	序論	1
1.1	研究の背景とその目的	1
1.2	従来の研究の概観	3
1.3	研究の特長	4
1.4	論文の構成	7
2	モバイルコンピューティング環境	9
2.1	まえがき	9
2.2	モバイル通信アーキテクチャ	10
2.3	ネットワーク環境	12
2.4	課題	13
2.4.1	移動性	14
2.4.2	切断性	16
2.4.3	通信サービス品質	17
2.5	あとがき	23
3	マルチメディア通信方式	24
3.1	まえがき	24
3.2	同期・非同期通信方式	24
3.2.1	まえがき	24
3.2.2	背景	26
3.2.3	同期・非同期通信方式	28
3.2.4	会議形態の切り替え	32
3.2.5	会議情報の蓄積	34
3.2.6	実装と評価	35
3.2.7	あとがき	39
3.3	選択的マルチメディア通信方式と先読み方式	39
3.3.1	まえがき	39
3.3.2	関連研究	41
3.3.3	選択的マルチメディア通信方式と先読み方式	42
3.3.4	実装	45

3.3.5	性能評価	49
3.3.6	あとがき	55
3.4	実時間指向優先転送方式	56
3.4.1	まえがき	56
3.4.2	内外の研究状況	57
3.4.3	ネットワーク応答特性プロトコル	62
3.4.4	マルチメディア通信システム適用例	70
3.4.5	シミュレーションによる性能評価	71
3.4.6	BSD/OS への実装	76
3.4.7	あとがき	78
3.5	あとがき	80
4	モバイル電子会議システムへの適用	82
4.1	まえがき	82
4.2	モバイル電子会議システム	82
4.2.1	要件	83
4.2.2	マルチメディア通信方式の役割	84
4.3	マルチメディア通信システム開発ツール	87
4.3.1	実時間プロトコル設計言語	88
4.3.2	実時間プロトコルシミュレータ	92
4.3.3	同期・非同期通信方式への適用	94
4.3.4	実時間指向優先転送方式への適用	97
4.4	あとがき	104
5	結論	105
	謝辞	107
	参考文献	108
	筆者発表論文	115

第 1 章

序論

1.1 研究の背景とその目的

携帯電話、PHS などの広域無線通信インフラの発展 [1] と、ノートパソコン、PDA などの携帯端末の高機能化と低価格化による普及を背景に、これらを利用していつでもどこでも情報を発信、受信できるモバイルコンピューティングが注目を集めている [2]。携帯電話のユーザは 3000 万人、PHS のユーザは 700 万人（平成 10 年初頭）と 3,4 人に一人がセルラ電話を持つようになり、現在のところ音声利用がほとんどであるものの、少しずつ電子メール、WWW ホームページ閲覧などのデータ通信利用が増えてきている。携帯電話は PDC による 9.6Kbps の回線交換型データ通信と 28.8Kbps のパケット通信、PHS は PIAFS による 32Kbps の回線交換型データ通信を行うことができ、画像のやりとりやテレビ電話などのマルチメディア通信を実現する基盤ができた。携帯端末に関しては、サイズ、重量を小さくしつつ [3]、CPU はより高速に、メモリ、ストレージはより容量を増やす方向にあり、マルチメディア情報を扱うプラットフォームとしての能力を持つようになった。

モバイルコンピューティング、モバイルコミュニケーションの大きな特長は、なにか伝えたい情報が発生した時、なにか情報が欲しくなった時、迅速にその要求を処理できることである [4][5]。株価、渋滞情報、チケット予約など鮮度が重要な情報を取得する場合、情報が欲しい時に即時に取得する必要がある。また、在庫管理、アポイントメント管理などは即時に情報を発信、処理しておくことで、後で情報を処理するよりも正確性を上げ、忘れてしまうこともなくなる。

モバイルコンピューティングにおいて、マルチメディア情報を扱いたいという要求が、ビジネス、エンターテイメント、公共サービス、それぞれに存在する。主に、蓄積コンテンツ取得とヒューマンコミュニケーションの 2 つのアプリケーションが考えられる。蓄積コンテンツ取得の事例として、出先でプレゼンテーションを行うためにビデオクリップをオフィスのパソコンから取り出したり、電車の中でニュースのビデオクリップを鑑賞したり、車中でカーナビを使って付近の観光情報を呼び出したり、あるいは美術館で関連する映像データベースからデータを取得するといった例が挙げられる。ヒューマンコミュニケーションに関しては、ビデオ電話、協同作業を伴うビデオ電子会議が考えられる。

一方、インターネットの急速な普及と共に、モバイルコンピューティング、モバイル

コミュニケーションはインターネット、イントラネットと密接に連携することが必須となっている。イントラネットと連携して、電子メールやグループウェアを利用したいという要求はもちろんのこと、他にもいくつかの理由がある。第一に、ユーザは出先や移動中にどんな通信手段、メディアを利用しようとも、とにかくインターネットにアクセスすれば、情報ソース、あるいは情報の宛先にアクセスできるという自由度を得ることができる。つまり、ユーザは品質、コスト、使い易さにより通信手段、メディアを選ぶことができる。第二に、WWW ホームページが爆発的に増えているように、インターネット上で有用なコンテンツが次々と提供されており、インターネットから情報を受信するという形態が増えている。第三に、インターネットは個人が複数の人、あるいは大勢に発信したいという要求にも応えることができる。

しかし、広域無線通信ネットワークとインターネットの連携した利用環境においてマルチメディア情報を扱うには様々な問題がある。ビデオ、オーディオのような実時間制約を持つ連続データの通信を行う場合、通信路は連続データが要求する通信サービス品質（データ転送速度、誤り率、遅延など）を保証しなければならない。しかし、無線通信には低速、不安定、スループットや遅延の変動といった通信サービス品質の課題がある。インターネットもベストエフォートという概念に基づき、基本的に品質保証の仕組みを持たないので、輻輳や負荷の増大により通信サービス品質は低下する。これまでのインターネットのトラフィックは、電子メール、ファイル転送、ニュース、遠隔ログイン、WWW など品質保証や実時間制約を必要としないデータ通信であったが、近年、CU-SeeMe に代表されるビデオ会議、RealVideo に代表されるビデオオンデマンドなどのマルチメディアアプリケーションの普及と共に、ビデオ、オーディオ、ボイスのような実時間トラフィックが急増してきている。

主にこの通信サービス品質の問題が、現在のモバイルコンピューティング環境でやりとりされるマルチメディア情報を制限している。低品質なオーディオ、イメージ（静止画）中心で、高品質なオーディオやビデオ（動画）が利用されることは少ない。

さらに、モバイルコンピューティング特有の問題として、携帯端末の移動性、切断性、稼働性を考慮しなくてはならない。セルラシステムはハンドオーバーと呼ばれる技術により、セルをまたがった移動をサポートする。これに加え、インターネット上の移動をサポートする必要がある。携帯端末は常に固定のアクセスポイントからインターネットに接続するとは限らず、移動して別のアクセスポイントからインターネットに接続するかもしれないからである。また、通信路の状態（電波状態、高速移動、サービスエリアの制限）、回線接続の場合は通信コスト、端末のバッテリーの制限などから、携帯端末はインターネットに常時接続している状態ではなく、間欠的にインターネットへの接続と切断を繰り返す。ユーザが移動したり、切断しても確実、円滑に情報のやりとり、コミュニケーションを行うことを可能にするような移動と切断を前提としたシステムを設計を行う必要がある。

本研究は、ユーザが携帯端末と広域無線通信機器を利用して蓄積マルチメディアコンテンツの取得や、マルチメディア情報を使ったヒューマンコミュニケーションを行うことを目的にしている。広域無線通信とインターネットが連携し、ユーザが移動、切断するというネットワーク環境において、インターネットの枠組み、すなわちネットワーク層以上で、モーバ

イルコンピューティング特有の課題、通信サービス品質の課題を議論する。ユーザの切断の課題を解決するために同期・非同期通信方式を、通信サービス品質の課題を解決するために選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式を提案し、実装、実験を行い、有効性を確かめた。また、これらの通信方式を組み合わせることでモバイルマルチメディア通信システムを構築できることを確かめたので報告する。

1.2 従来の研究の概観

モバイルコンピューティング環境におけるマルチメディア通信に関する技術項目は、広域無線通信のためのマルチメディア通信技術、インターネットのためのマルチメディア通信技術、モバイルコンピューティングのサポート技術の3つに大きく分けられる。

広域無線通信のためのマルチメディア通信技術 広域無線通信、特に移動通信においては雑音、干渉、フェージングなどにより誤り率の増加、スループットの低下、遅延の増大といった通信サービスの品質が低下する [6]。そこで、誤りに強く、低ビットレートで、メディアスケラビリティを提供する圧縮、符号化方式や、前方向誤り訂正 (FEC:Forward Error Correction), 自動再送要求 (ARQ:Automatic Repeat Request) などの誤り制御技術を用いたビデオ転送などが検討されている [7][8][9]。

圧縮符号化方式に関しては ITU-T 勧告の低ビットレートのビデオ会議用映像符号化方式 H.263 [10] や、ボイス符号化方式 G.729、人物以外の背景部分の映像を低精細、低フレームレートにし、消費帯域幅を削減する技術 [11] などの技術を利用できる。ビデオ内の人間や車、建物などのオブジェクトを検出し、ビデオを各オブジェクトの集まりとしてみなすコンテンツベースの圧縮符号化方式は 1/10000 の圧縮を達成すると共に、柔軟なスケラビリティを提供する [12]。符号化する際にビデオの品質を調節することでデータ量を増減させるような可変ビットレート符号化方式もある [8]。

無線を使って WWW サーバから取得する際に、画像の品質、容量を落として無線に流す研究もある [13]。広域無線通信は低速、不安定であり、大容量の画像の取得にかなりの時間がかかり、ユーザにストレスを強いる。そこで、取得する GIF 画像を JPEG に変換したり、JPEG の Q ファクタを落として画像の容量を減らす。

インターネットのためのマルチメディア通信技術 インターネットのようなパケット交換ネットワークにおいてビデオ、オーディオ転送を行うには、エンドツーエンドで一定以上のスループットが達成され、各パケットの遅延の変動 (ジッタ) が小さくなければならない。インターネットは様々なノード、ネットワークを含む異機種分散環境で、特別な品質制御を行わないベストエフォートサービスに基づいている。経路上のあるノードでトラフィックが増加し、負荷が上がると輻輳が起これ、スループットは低下し、パケットの消失、パケットの遅延の増加などが起こる。

経路上のすべてのノード、ネットワークが通信サービスを保証する制御機能を持てば、通信サービスを保証することができる。すべてのパケット交換のノードが RSVP (RFC2205) [14]

のような資源予約プロトコルに対応し、CBQ[15]やWFQ[16], FIFO+[17]などのキューイング方式、許諾制御を実装しなければならない。

そこで、アプリケーションがマルチメディア情報の空間的、時間的解像度を調節してスループットの変動に追従できるように消費帯域幅を削減したり、バッファリングや再生点の調節などを行って、パケットの転送遅延の変動を吸収するような適応的手法が提案されている。ビデオの再生側から送信側に達成されたフレームレートをフィードバックし、送信側が送信フレームレートを調節することにより、消費帯域幅やCPU処理量を減らすことができる[18][19][20]。IVS[21],vic[22]はH.261符号化方式を利用するインターネット上のビデオ会議システムで、動的に映像の出力データ量を制御し、スループットの変動に適応することができる。また、階層的符号化[23]の技術もネットワーク環境への適応に利用できる。

ビデオ、オーディオ、あるいは実時間データのトランスポートプロトコルとしてはRTP[24]が標準であり、各パケットの再生タイミングを示すタイムスタンプの機能などを持つ。

モバイルコンピューティングサポート技術 移動の問題に関しては、無線LANやPHS、携帯電話などのセルラシステムは独自にそのネットワーク内の移動を位置登録、ハンドオーバなどの技術でサポートする[25][6][26][27]。インターネットへのアクセスポイントが一定である限り、すなわちIPアドレスが同じ間欠的にインターネットへの接続と切断をである限り、無線LAN内やセルラシステムのセル間の移動はネットワーク層(IP層)よりも下層で透過的に行われる。

しかし、無線LANでもサブネットをまたがって移動したり、広域インターネットを移動するにはIP層での移動サポートが必要である。そこで、Mobile IPと呼ばれる技術がいくつか検討されてきた[28][29]。Mobile IPのいくつかの実験的に実装が報告されている。現在は移動先でDHCP(Dynamic Host Configuration Protocol[RFC1541]) [30]を使ってIPアドレスを取得する方法が一般的である。

切断に関しては、送受信中にユーザが自発的に切断したり、意図しない切断が起きても再接続時に途中から転送を開始したり、自動的に再接続するようなモバイルミドルウェア[31][32]、エージェント[13]、筆者らが開発してきたモバイルソケット[33]などの技術が提案されている。さらに、モバイルホストがネットワークから切断中であってもキャッシングによりファイルアクセスを可能にするファイルシステム[34]、複数のホストが並行に更新を行った場合の衝突の検出、解決手法[35][36]などの研究がある。

1.3 研究の特長

本研究全体の特長 本研究の主要な特長はモバイルコンピューティング環境におけるマルチメディア通信方式を扱っている点である。広域無線通信ネットワークとインターネット、イントラネットが連携したネットワーク環境を想定し、端末の移動や切断を考慮する。これまでもマルチメディア通信について多くの研究がなされてきたが、それはインターネット、あるいは無線通信のどちらかを対象としたものであった。また、端末の切断を考慮した研究

もほとんどなかった。

本研究の提案する同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式は、異機種分散環境を考慮したアプリケーション層におけるマルチメディア通信方式を提案しており、有線、無線の違い、通信メディア、ネットワーク、OSの種類に依存せず、広域無線通信を含むインターネット、イントラネットに広範囲に適用可能である。実時間指向優先転送方式は、インターネットのIP層のキューイング方式を変更するものであるが、IPと互換性を持っていて、経路上の一部のノードが実時間指向優先転送方式に対応していれば効果を発揮できるという長所があり、やはり広範囲に適用可能である。

また、提案通信方式の仕様、動作を実時間プロトコル設計言語 *T - PROMELA* を用いて記述し、言語専用のシミュレータ *T - SPIN* によって動作確認、性能評価を行った点も本研究の特長である。

以下で同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式、それぞれの特長について述べる。

同期・非同期通信方式の特長 同期・非同期通信方式の研究の特長は、ネットワーク、会議システムから切断する機会の多いモバイルコンピューティングを行っているユーザ（モバイルユーザと呼ぶ）の利用を考慮している点である。従来の通信方式は、ユーザ端末はオフィス、学校、家庭内のデスクトップコンピュータで、常に有線ネットワークにつながっていると仮定していることが多かった。しかし、モバイルユーザはユーザの都合や、通信路状態の悪化、端末バッテリーの節約、通信コストの削減などの理由から、ネットワークから切断する機会が多い。通信相手が機械の場合、切断の問題は、モバイルミドルウェアなどのモバイルコンピューティングサポート技術を利用することにより解決することができる。しかし、ヒューマンコミュニケーション（会議）の場合、切断によるコミュニケーションの中断は使い勝手を大きく損なう。

本方式の特長は、第一に、ユーザの要求や通信路の状態に従って、動的に会議形態を同期、非同期に切り替えることのできる自由度を提供することである。同期会議中にユーザが非同期の参加を要求したり、通信路が切断した場合、会議形態を非同期会議にする。会議形態に関係なく、ユーザは1つのアプリケーションを使うことができ、ホワイトボードなどの共有空間、蓄積されたコミュニケーション情報が引き継がれる。逆に非同期会議中にユーザが他のユーザを呼び出して、同期会議に切り替えることも可能である。

第二に、コミュニケーションでやりとやりとりされる情報が共有情報の場合、共有の会議サーバに蓄積する集中型の蓄積、個人的情報の場合、各ユーザ専用のホームサーバに蓄積する分散型の蓄積を行う点である。共有情報は、情報が更新されたときの一貫性をとるためとストレージの有効利用のために、共有の会議サーバに蓄積する。個人的な情報の場合、ユーザ自身が保持、管理し、自由に加工や削除を行えるように、ユーザの所有するホームサーバに蓄積する。

選択的マルチメディア通信方式の特長 選択的マルチメディア通信方式の特長は、コンテンツベースの適応的制御を行う点である。エンドツーエンドの適応的手法は、受信側でスループットの変動や達成フレームレートを監視して、現在の通信サービス品質状態を理解し、マルチメディア情報の空間的、時間的解像度を調節して、転送データ量を適切な大きさにする。エンドツーエンドの通信路の運搬能力を越えるトラフィックはネットワークリソースと送信側、受信側の CPU リソースの浪費である。適切な転送データ量に設定することで、結果的にユーザが知覚するメディアの再生品質を向上することができる。

しかし、従来の適応的手法はマルチメディア情報の中身を考慮せずにフレームレートや画像の色数、画像のピクセル数を落とすといったメディアスケール処理を行うため、ユーザが重要な情報を獲得し損なうことがあるという欠点があった。例えばゴルフやテニスなどのレッスンビデオをみる場合、スイング動作のような重要なシーンでフレームレートやピクセル数が減少しては意味がない。しかし、それ以外のインストラクターの解説のシーンのビデオはそれほど重要ではないだろう。

そこで、選択的マルチメディア通信方式はマルチメディア情報の作成者、あるいは提供者がマルチメディア情報のシーンごとに、その意味的重要度に基づいて優先度を与え、その優先度に基づいた選択的なマルチメディア転送を行う。あらかじめ優先度付けが必要なため、本方式は蓄積型のビデオを対象とする。ビデオは連続したシーンから構成され、シーンは連続したビデオフレームから構成される。ビデオフレームを転送する際に、低優先度が与えられたシーンは積極的にフレームレートを落とし、そのかわりに高優先度が与えられたシーンのフレームをまえて配送し、受信側はストレージに保持しておく。つまり、前後のシーン間で転送量の配分を行う。データ転送速度が不足している状況では、低優先度シーンのフレームレートの大幅な低下を被るものの、重要なシーンを確実に受信者に提供することができる。

先読み方式の特長 先読み方式の特長は、スループットの変動や比較的大きな遅延の変動が起きる通信環境を対象としたエンドツーエンドのマルチメディア通信方式を提案している点である。従来の手法も短い転送の途切れ、ある程度までの遅延の変動をバッファリングや再生を遅らせることで吸収することができたが、パケットの遅れや消失が連続したパケットで集中すると、バッファ内のフレーム数が極端に低下し、バッファのアンダーフローが起きる。

そこで、受信者は一部のビデオフレームを先読みし、転送が一時的に途切れても再生を続行することを可能にする。突然、転送が途切れてもストレージに先読みされたビデオフレームを再生することができる。また、先読みされたパケットがロスした場合の再送を提案する。インターネットの連続メディアの実時間通信では再送しても再生タイミングに間に合わないという理由から再送が行われることは少ないが、先読みされたパケットは再送する時間的余裕を持っている。再送により、ビデオ再生品質を向上できると考える。

実時間指向優先転送方式の特長 実時間指向優先転送方式の特長は、IP パケットに対し、優先度と実時間制約の両方を要求することができる点である。従来も優先度の指定は可能であったが、実時間制約を指定するのものはほとんどなかった。重要なパケットでも実時間制約が緩い場合もあるし、重要でないパケットでも実時間制約が厳しい場合がある。このように、パケットの重要度と実時間制約は独立であり、両方を要求できることは有用であると思われる。

従来、実時間通信に対して資源予約の手法が盛んに研究されてきたが、資源予約には通信前のオーバーヘッド、広域での利用可能性、利用効率などの問題がある。実時間指向優先転送方式は、ベストエフォート（最善努力型）の概念に基づいており、パケットの優先度と実時間優先に従ってパケットスケジューリングを行う。通信前のオーバーヘッドがなく、後述するように広域での利用可能性も考慮されている。パケットスケジューリングによる利用効率の低下が懸念されるものの、バーチャルサーキットの構築やフラグメントの回避により、性能低下を防ぐことができると思われる。

実時間指向優先転送方式は、現在インターネットで広く用いられている IP[37] と互換性を持つネットワークプロトコル NPP を導入する。NPP はインターネットのエンドエンドのノードとその中間のノードに配備される。インターネット環境の場合、経路上のすべてのホスト、ルータに提案プロトコルが実装されていることを仮定するのは難しいが、NPP はすべてのノードに配備されなくても動作するように、広域での利用可能性を考慮して設計されている。

1.4 論文の構成

本章ではモバイルコンピューティング環境におけるマルチメディア通信方式に関する従来の研究を概観し、本研究の特長を述べた。

2章では、研究の背景として、モバイルコンピューティングのサポート技術について説明し、対象とするネットワーク環境や研究課題について明らかにする。まず、2.2でモバイル通信アーキテクチャについて紹介する。イントラネット内に携帯端末の代理としてサーバと通信を行うエージェントや、切断の多い通信環境において確実、かつ効率的な通信を行うための蓄積交換を行うモバイルミドルウェア、非同期操作（切断時操作）に対応するアプリケーション等を導入することにより、低速で不安定で、通信コストの高い無線通信におけるモバイルコンピューティングを実用的なものにする。

2.3では、本論文が対象とするネットワーク環境を示す。本論文はイントラネットを含む広域インターネットと広域無線通信が連携するネットワーク環境を想定している。ユーザがインターネットを移動し、インターネットから頻繁に切断する。2.4では、このネットワーク環境におけるマルチメディア通信の課題は、移動性、切断性、通信サービス品質の3つであることを述べ、移動性については DHCP, Mobile IP, 無線通信システムの移動サポートなどの既存の技術で解決できることを示す。

3章は2.4で述べた残る課題、すなわち切断性の課題、通信サービス品質を解決するマル

チメディア通信方式を提案する。同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式という4つのマルチメディア通信方式を提案する。

3.2では、ビデオ会議システムやビデオ電話などにおけるヒューマンコミュニケーション(会議)が、ユーザの都合や通信の切断により中断されるという問題を改善する同期・非同期通信方式を提案する。3.2.3で同期通信方式のモジュールやシステム構成、会議の開始手順について述べ、3.2.4で同期・非同期・部分非同期の会議形態の切り替えの手順、3.2.5で集中型、分散型の会議情報の蓄積の仕方について論じる。最後に3.2.6で実装と会議実験の結果について報告する。

3.3では、低速で、スループットが変動するという広域無線通信ネットワークとインターネット環境において、ユーザがインターネット上の蓄積型のマルチメディアコンテンツに効率的にアクセスすることを可能にする選択的マルチメディア通信方式と先読み方式を扱う。3.3.3で選択的マルチメディア通信方式 SMAP と先読み方式を提案の提案を行い、ビデオフレーム、オーディオブロックを転送するスケジューリングアルゴリズムについて述べる。3.3.4では両方式を組み込んだサーバ、ミドルウェア、アプリケーションの実装について述べる。3.3.5では両方式の性能評価について報告する。

3.4では実時間指向優先転送方式について論じる。3.4.3で基本的なアイデア、プロトコル階層、プロトコルの規定、パケットスケジューリングのアルゴリズム等について述べる。3.4.4では実時間指向優先転送方式のマルチメディア通信システムへの適用例について述べ、3.4.5でシミュレーションによる性能評価を行う。3.4.6で BSD/OS 2.0 への実装について述べる。

4章ではマルチメディア通信方式を実際のアプリケーションでどのように利用するか、そしてどのように構築するかについて述べる。4.2で、マルチメディア通信方式の適用例として、モバイル電子会議システムを取り上げ、同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式の利用について述べる。各マルチメディア通信方式を組み合わせたときの効果や問題点に関して考察する。

4.3で、マルチメディアアプリケーションを開発するためのツールとして、実時間プロトコル設計言語とそのシミュレータについて紹介する。言語の実時間記述能力やシミュレータの動作原理を述べ、実際に同期・非同期通信方式の会議形態の切り替え、実時間指向優先転送方式の設計仕様を実時間プロトコル設計言語を用いて記述する。そして、シミュレータにより動作確認と性能評価を行う。

最後に5章で結論として本研究の成果をまとめ、今後に残された課題について述べる。

第 2 章

モバイルコンピューティング環境

2.1 まえがき

現在のモバイルコンピューティングは、イントラネットに接続してメールサーバ、WWWサーバ、データベースサーバにアクセスして電子メール、ホームページ、あるいはデータ、ファイルを取得するという形態が中心である。PHS や携帯電話などの広域無線通信インフラの発展 [1] から、特に最近、広域無線通信を使ってインターネット、イントラネットにアクセスするワイヤレスモバイルコンピューティングが注目を集めている。公衆電話を探したり、出先の電話を借りることなく、無線通信のサービスエリア内であれば、利用可能であるという自由度の高さがある。

しかし、広域無線通信には、低速で、不安定で、通信コストが高いという問題がある。本章では、エージェント、モバイルアプリケーション、モバイルミドルウェアから構成されるモバイル通信アーキテクチャを紹介する。エージェントを介した通信を行うことで、携帯端末とサーバ間の通信量、通信コストを減らす。モバイルミドルウェアが通信の途中で切断しても再接続の際には途中から通信を再開できるような機構を提供する。モバイルアプリケーションはネットワークから切断している状態でも、それを隠して動作する。

一方、現在、インターネット上ではビデオ会議、電話、ラジオ、ビデオ放送、ライブ中継などのマルチメディア通信アプリケーション、システムが次々に登場し、ユーザを増やしている。例えば、CU-SeeMe¹、Microsoft NetMeeting²はインターネット上のビデオ会議、電子会議、RealAudio、RealVideo³はラジオ放送、ビデオ放送、NVAT(Network Video Audio Tool)はライブ中継を実現する。マルチメディアアプリケーションの多くは、蓄積コンテンツの取得とビデオ会議、ビデオ電話の2種類のカテゴリーに分けることができる。

本章では、これらのマルチメディアアプリケーションを広域無線通信とインターネットが連携するネットワーク環境で利用する際の課題を明らかにする。

¹<http://cu-seeme.cornell.edu>

²<http://www.microsoft.com/netmeeting/>

³<http://www.real.com/>

種類	広域移動無線通信	LAN	インターネット
転送速度	9.6K ~ 29.2K	10M,100M ~	多様
遅延	大きい	小さい	多様
スループット	変動	高スループット	変動
切断	切断あり	なし	なし
通信コスト	高い	無料	無料~安価

表 2.1: 広域移動無線通信 vs イントラネット LAN(vs インターネット)

2.2 モーバイル通信アーキテクチャ

携帯端末と広域無線通信を使って、例えばイントラネット内のデータベースやファイル、メール、あるいは WWW のホームページにアクセスするようなモバイルコンピューティングを行うとする。広域無線通信とイントラネットの LAN では転送速度、遅延、安定性、通信料金が全く異なり(表 2.1)、従来の LAN 上のシステムをそのまま広域移動無線通信上で利用すると、応答性や効率、切断によるやり直しなどの問題が浮上する。低速な無線通信で大量のデータを送ることは難しいので、転送データ量を減らす必要がある。また、無線通信が突然切断した場合、再接続して確実に通信を完了させなければならない。通信コストが高いので、できるだけ短時間で通信を終えたいという要求もある。

通信量、切断、通信コストの問題に対し、図 2.1 のようにエージェント、モバイルアプリケーション、モバイルミドルウェアを含むモバイル通信アーキテクチャが登場した。

LAN 上にエージェントを配置し、携帯端末はエージェントに要求を投げてネットワークから切断する。エージェントはデータベースサーバやファイルサーバや、WWW サーバにアクセスし、要求された仕事を代理で実行する。エージェントは携帯端末に応答を返す準備ができたなら、携帯端末を呼び出し(あるいは携帯端末が適当な時間待ってから再びネットワークに接続し)、携帯端末はエージェントから結果を受け取る。エージェント・サーバ間の LAN を活用し、クライアント・エージェント間の通信を減らすことで、無線通信の低速の問題、通信コストの問題を回避する。また、WWW 上の画像を取得する際にエージェントが画像の品質を落とし、低速な無線で扱いやすいようにデータ容量を落とすこともできる [13][38]。

また、アプリケーション層とトランスポート層の間にモバイルミドルウェアが配置される。モバイルミドルウェアは不安定なリンクにおいてもデータ通信を保証し、データ転送中に切断が起きて再接続した場合は途中から転送を再開し、切断中でもメッセージ通信を行うことを可能にする蓄積交換の機能を提供する [31][33]。Mobile Component(NTT データ) [32][39]、Oracle Mobile Agents(Oracle)、SQL Remote (サイベース) ⁴、

⁴<http://www.sybase.co.jp/wp/REMOTE/>

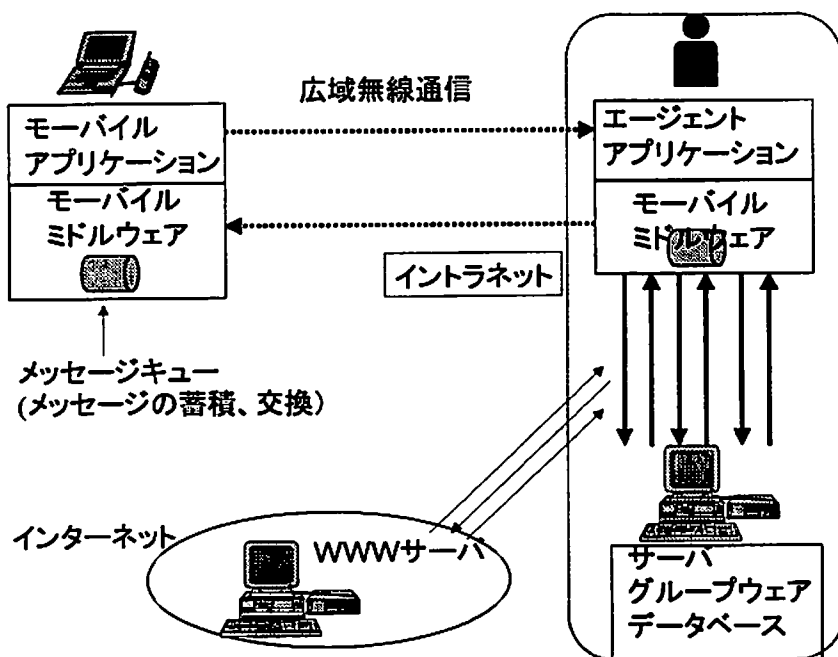


図 2.1: モバイル通信アーキテクチャ

IBM ARTour(IBM)⁵ ⁶などのモバイルミドルウェア製品が提供されている。

さらに、携帯端末は常時ネットワークに接続しているわけではない。モバイルアプリケーションは非同期動作に対応し、ネットワークから切断していても動作する。グループウェアの Lotus Notes R4J(ロータス)⁷は自宅や出張先でも電話回線を通してサーバに接続して、必要な情報を携帯端末に複製することで(レプリケーション機能と呼ばれる)、ネットワークから切断しても社内とほぼ同一の条件で作業することを可能にする。

また、CODA ファイルシステム [34] は、あらかじめ必要だと予測されるファイルをキャッシングしておくことにより、携帯端末がネットワークから切断中でもファイルアクセスを可能にする。ただし、接続時に複数の書き込みによる衝突を検出、解決することが必要になる [35]。

他にも RAS(Remote Access Server) にエージェントを配置し、携帯端末間で無線専用の軽量プロトコルを用いるような通信アーキテクチャも提案されている [40][41]。エージェントが RAS 上で無線専用の軽量プロトコルと TCP/IP のプロトコル変換を行う。TCP/IP のプロトコルヘッダのオーバーヘッドによる性能低下、無線の不安定さによる不必要な輻輳制御に起因する TCP/IP の性能低下などを防ぐことができる。

⁵<http://www.software.ibm.com/sw-sell/oem-sw/oemproducts/network/artour.html>

⁶<http://www.ibm.co.jp/manuals/gc31-8414-00/>

⁷<http://www3.lotus.co.jp/productinfo.nsf/NotesR4J?openview>

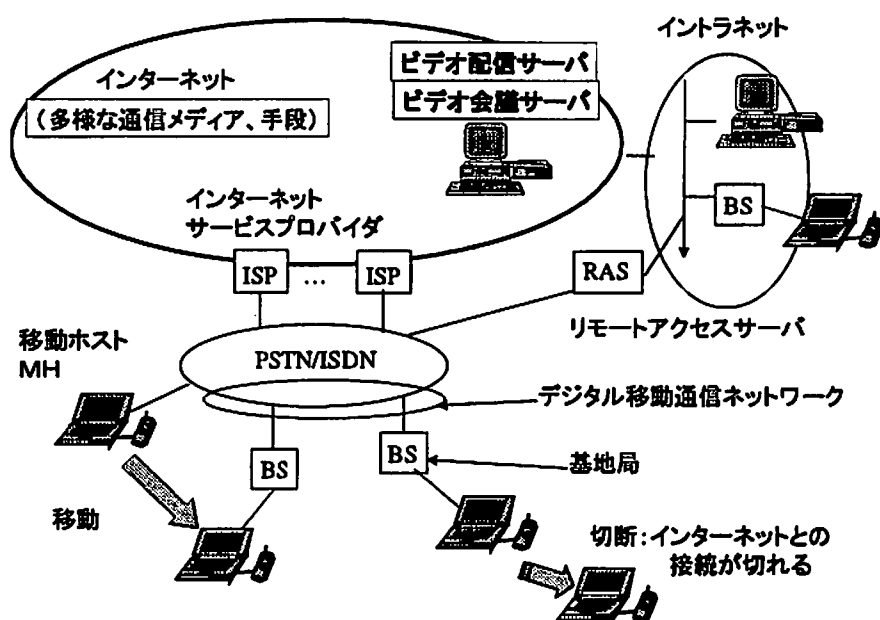


図 2.2: ネットワーク環境

2.3 ネットワーク環境

2.2のモバイル通信アーキテクチャは、イントラネットと広域無線通信の連携したネットワーク環境を想定している。本論文は図 2.2のような広域無線通信とインターネット、イントラネットの連携しているネットワーク構成を想定している。モバイルユーザの端末 MH(Mobile Host)は移動中や出先において、PHS、携帯電話を介してインターネットサービスプロバイダ (ISP) や、属する組織のリモートアクセスサーバ (RAS) に接続し、インターネット、イントラネットにアクセスする。ユーザは移動中や出先において公衆電話を探したり、出先の電話を借りることなく、必要なときに通信することができる (ただし、無線通信のサービスエリア内である必要がある)。

そして、MHはインターネット上、イントラネット内のビデオ配信サーバ、ビデオ会議サーバに接続し、ビデオ配信サーバから蓄積コンテンツを取得したり、ビデオ会議サーバ上で協同作業を行ったり、ビデオ電話で会話する。

携帯電話 (PDC, PDC-P) に比べ高速で通信コストが安く、マルチメディア通信の利用が期待される PHS の PIAFS を利用する場合のシステム構成を図 2.3に示す。PIAFS[42] は、PHS を利用したマルチメディア通信の普及を促進することを目的として設立された団体である PHS インターネット・アクセス・フォーラム⁸によって策定されたエンドツーエンドの伝送制御 (データリンク層) プロトコルである。簡易な再送制御 (MODS-ARQ 方式) でスループットが高い (最高 29.2Kbps)、柔軟な上位プロトコルが選択可能、スループット低下のない全二重通信が可能といった特徴がある。PIAFS は移動端末側では PIAFS 対応の PC

⁸ <http://www.infopro.or.jp/piaf/j/aboutpiaf.html>

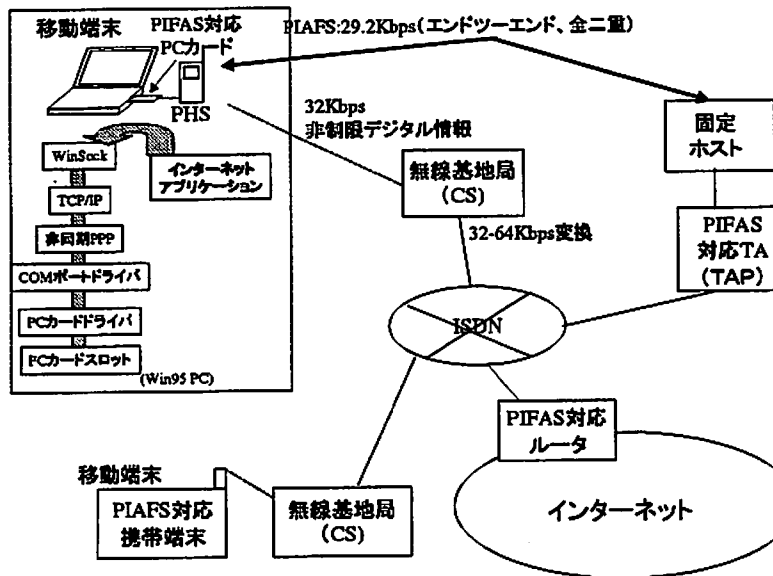


図 2.3: PIAFS を利用した PHS データ通信

カードアダプタ、PIAFS 内蔵型の携帯端末、固定端末側では PIAFS 対応ターミナルアダプタ (TAP) や PIAFS 対応ルータによりサポートされる。現在、多くのプロバイダが PIAFS に対応したアクセスポイントを提供している。また、一部の PHS サービス会社が 64Kbps のデータ通信サービスの予定をアナウンスしている。PIAF は動画に関して、専用のリアルタイムプロトコルを策定中である。

2.4 課題

2.3で述べたネットワーク環境でマルチメディア情報をやりとりするには、以下のような3つの解決すべき課題がある。

1. 移動性の課題：移動端末への送信、移動端末の受信

モバイルコンピューティング環境では、ユーザが接続するインターネットのアクセスポイントは変わる可能性を考慮しなければならない。出先では、ユーザは通信コストを削減するために、イントラネットやいつもの自宅近くの ISP のアクセスポイントでなく、最寄りの ISP のアクセスポイントに接続することを選ぶかもしれない。そこで、自端末が移動してもマルチメディア情報を確実に受信できる技術、相手端末が移動しても確実に送信できる技術が必要である。ビデオ、オーディオ通信中に移動する場合は通信が途切れないことも要求される。

2. 切断性の課題：通信の途中再開、円滑なヒューマンコミュニケーションの阻害、切断端末への送信

無線通信が不安定で突然接続が切れることがあること、通信コストが高く、バッテリーに制約があるのでユーザがこまめに通信の接続、切断を行うことから、MH は間欠的にネットワークに接続する。画像取得中に切断したときに画像データの最初から転送するのではなく、残りを途中から取得するような通信の再開技術、切断したら自動的に再接続するような技術が必要である。さらに、協同作業やビデオ電話など、通信相手が人間の場合、切断は円滑なヒューマンコミュニケーションを阻害する可能性がある。他にも、相手端末が切断している場合に、送信をどうするかという問題もある。

3. 通信サービス品質の課題：インターネット、広域無線通信のような通信サービス品質が保証されない通信路上のマルチメディア通信
ビデオ、オーディオのような実時間制約を持つ連続データの通信を行う場合、通信路は連続データが要求する通信サービス品質（データ転送速度、誤り率、遅延など）を保証しなければならない。保証がなされない場合、上位層で提供される通信サービス品質に適応する仕組みが必要である。

以降の節で、これらの各課題について詳しく見ていき、本論文のその課題に対するアプローチについて述べる。

2.4.1 移動性

無線 LAN や PHS、携帯電話などのセルラシステムはハンドオーバーなどの技術を使って、自ネットワーク内での移動をサポートする [25][6][26][27]。インターネットへのアクセスポイント) が一定である限り、すなわち IP アドレスが同じである限り、無線 LAN 内やセルラシステムのセル間の移動は IP 層よりも下層で透過的に行われる。

これまでのインターネットは、ホストがインターネット内を移動することを全く考慮して、ホストが固定的に接続されていることを前提に、プロトコル、アプリケーションが設計されてきた。しかし、コンピュータの小型化により、ネットワークから移動して別のネットワークに接続するという利用形態がでてきた。IP アドレスはネットワーク番号、サブネット番号、ホストの番号から構成されており、ホストの識別子というよりも、そのホストのインターネット上の位置を示している。無線 LAN において別のサブネットに移動したり、社内のサブネットに接続している携帯端末を切断して、出先から近くのインターネットサービスプロバイダに接続する場合、ネットワーク上の位置の変化に伴い、IP アドレスを変える必要がある。

このとき、アドレスの割り当て、システム環境情報の提供が主に問題となる。これまで、新しいホストをネットワークに接続する場合、ネットワーク管理者が手作業で未使用のアドレス群から適当なものを割り当てて、メールサーバや DNS サーバ、プリンタサーバなどの環境情報を教えるというような手順を踏んでいた。移動ホストが増えれば明らかにこの手法は破綻する。

DHCP(RFC1541)[30] は、クライアント・サーバ型のシステム構成で、サーバは各種のネットワークコンフィギュレーション情報を保持しており、クライアントがあるサブネットに接

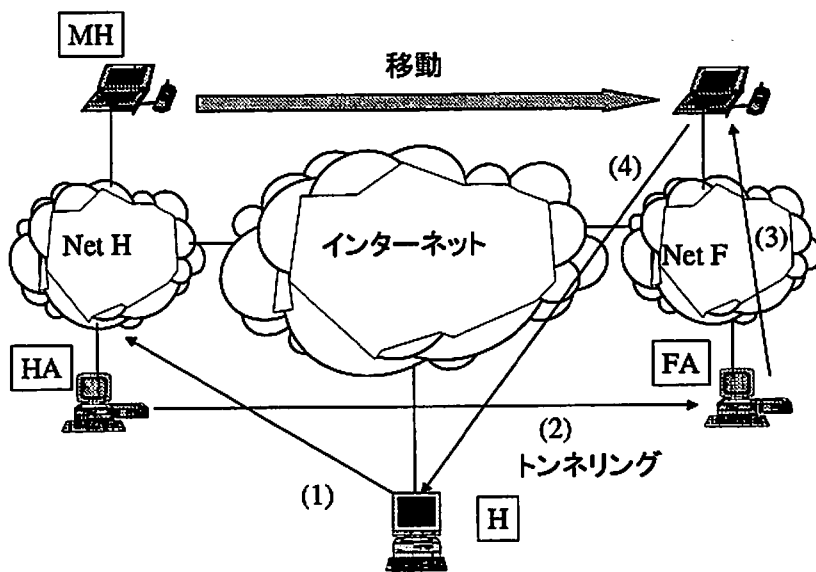


図 2.4: Mobile IP

続したとき、DHCP サーバに要求メッセージを送る。サーバは、(永続的、あるいは一時的に利用できる) IP アドレスとそのサブネットの環境情報をクライアントに与える。クライアントはその応答に基づき、ネットワークインタフェースの初期化を行うことができる。DHCP のクライアントは移動先のネットワークから IP アドレスを取得するので、移動先により IP アドレスが変わる。

コネクション型のトランスポートプロトコルである TCP は自分と相手の IP アドレスとポート番号の対でコネクションを管理しているので、IP アドレスが変わるとコネクションを維持できないという問題がある。サブネットを移動する度に、セッションが途切れるのでは使いにくい。そこで、ホストの移動透過性を提供するために、分割サブネット方式、ソースルーティング方式、識別子アドレスマッピング方式 (WIDE VIP), パケット転送方式 (IETF Mobile IP) などの技術が開発された [29]。

IETF で標準化されている Mobile IP [28] の動作を図 2.4 に示す。IETF Mobile IP は既存のネットワークに対する修正や設備の追加を少なく、移動ホスト MH (Mobile Host) を収容できる方式である。MH が普段、接続されている自宅のネットワークやオフィスのネットワークは「Home Network」と呼ばれ、MH の Home Network は Net H であるとする。MH は Net H に対応する IP アドレスを所有し、Net H に接続されている限り、通常の他のコンピュータと同様に通信を行なうことができる。ここで、MH が Net F に移動したとする。MH はまず移動先で「FA (Foreign Agent)」と呼ばれるコンピュータを探す。FA は MH をサービスするコンピュータであり、MH は FA に自分が何者かを伝え、そのネットワークを利用して良いという許可を得る。その後、FA は MH の場所を「HA (Home Agent)」に知らせる。HA は MH の Home Network に存在し、MH の居場所を管理、記

憶するコンピュータである。

ホスト H が MH と通信する場合の実際の通信手順は図の (1) から (4) のようになる。H は MH がどこにいるか知らないので、とにかく Net H にいると仮定してデータを送信する (1)。しかし、Net H に MH はいないので、代わりに HA が MH 宛のデータを受け取り、HA は FA との仮想回線を利用して (トンネリング)、データを FA に転送する (2)。FA は受け取ったデータを Net F に送り出すと、MH がそれを受け取る (3)。MH から CN へは、通常の送信手続きである (4)。

ただし、この方式には H から MH へいく通信はいったん HA-FA を経由するので、経路が冗長であるという問題がある。極端な例をいうと、Net H が日本にあって、Net F が米国にある場合、米国内の H と MH が通信しようとする、毎回 H から MH への通信はいったん日本を経由するという非効率な通信になる。現在、IETF で最適な経路で配送するための検討がなされている。Mobile IP は実装例がいくつか報告されおり (例えば、カーネギメロン大学から FreeBSD 等の BSD 系の OS で動作するバージョンが入手可能⁹)、これから広く使われることが予想される。

IETF Mobile IP により、自端末、相手端末が移動しても、マルチメディア情報を確実に送受信することができる。しかし、ビデオ、オーディオ通信中に移動する場合、通信が途切れないことが要求される。無線 LAN やセルラシステムでは独自にソフトハンドオフ、高速ハンドオフ技術により、移動して基地局が切り替わってもデータ転送が途切れないが [25][1]、インターネットの IP 層レベルでは、まだこの種の研究が少ない。

本論文では、広域無線通信を利用してインターネットにアクセスするような利用形態を考えている。この場合、インターネット接続中に IP アドレスが変わることはなく、広域無線通信システム独自の移動サポート技術により途切れずにデータを配送することができる。移動ホストは最寄りの、あるいは都合の良い ISP や組織のイントラネットに広域無線通信により接続し、DHCP により IP アドレスを割り当ててもらおう。移動中、その IP アドレスは変わらない。

以上のように、本論文の対象とする利用形態においては、移動の課題は広域無線通信システム、DHCP、Mobile IP を組み合わせて利用することで解決できると考える。

2.4.2 切断性

無線通信でインターネットに接続する携帯端末は以下のような理由からネットワークから切断する機会が多く、間欠的に接続状態になる。

- 無線通信が不安定：高速移動や無線通信のサービスエリア外に移動して無線通信が切断する
- 無線通信コスト：パケット交換で料金が従量制の場合、常時接続可能だが、回線交換の場合、利用時間で課金されるので利用者はこまめに切断する。

⁹ <http://www.monarch.cs.cmu.edu/software.html>

- バッテリー：端末がバッテリー駆動の場合、バッテリーが切れれば通信から切断せざるを得ない。小電力消費のスリープモードを備えていればバッテリーの持つ時間を増やすことができるが、駆動時間には限界がある。

蓄積コンテンツを取得する場合、切断性の課題は2.2で述べたモバイル通信アーキテクチャ、モバイルミドルウェアにより解決されることができる。例えば、画像取得中に切断したときに、途中まで取得したデータをもう一度受信するような非効率なことを避け、残りを途中から取得したり、切断したら自動的に接続することができる。

しかし、協同作業やビデオ電話など、通信相手が人間の場合、モバイル通信アーキテクチャ、モバイルミドルウェアだけでは不十分である。切断により、協同作業やビデオ電話による個人的コミュニケーションが中断されたりすると、円滑なヒューマンコミュニケーションが阻害される。モバイルコンピューティング環境では通信路の状態による切断だけでなく、ユーザが電車を乗り換えたり、歩いて移動しているときには協同作業には参加できないといったユーザの都合によるヒューマンコミュニケーションの中断も考えられる。

本論文はこの問題に対し、同期・非同期通信方式を提案する。ヒューマンコミュニケーションにおいて、全参加者が時間を共有してリアルタイムに参加する「同期会議」と各参加者が都合の良い時間に間欠的に参加する「非同期会議」を統合し、参加者が都合のよい会議形態で協同作業や個人的コミュニケーションをとることができるような柔軟性を提供する。また、一部の参加者が一時的に切断しているとき、その間のコミュニケーションの情報をストレージに蓄積することで、一部の参加者が不在でも他の参加者は会議を続行でき、不在だった参加者は後から蓄積情報を参照して会議進行の詳細を見ることができるような、切断を前提としたシステムを提案する。

さらに、相手端末が切断しているときに送信をどうするかという問題に対して、同期・非同期通信方式は、ユーザのホームネットワークにそのユーザを代理するホームサーバを配置する。これは2.4.1で紹介したHA(Home Agent)と似たもので常時インターネット上に存在し、そのユーザへのアクセスポイントとなる。HAは蓄積機能を持たないのに対し、ホームサーバは切断している端末の代わりにマルチメディア情報を受信し、蓄積する点が特長である。つまり、留守番電話のような役割をする。

3.2で同期・非同期通信方式について詳しく議論する。

2.4.3 通信サービス品質

連続メディアの特徴

ビデオ会議、電話などのライブ・マルチメディア通信は、常に情報が発生し、それをリアルタイムに配送する必要がある。また、WWW上で提供されているビデオ、オーディオなどの蓄積型のコンテンツもリアルタイムに取得したいという要求がある。ファイル転送プロトコルftp(File Transfer Protocol)、http(HyperText Transfer Protocol)によってビデオファイルをすべてユーザ側の端末に取得し終えてから、ビデオ再生を始めるという手順では、ユーザは読み込みが完了するまで再生を待たなければならず、ビデオ全体を保持するた

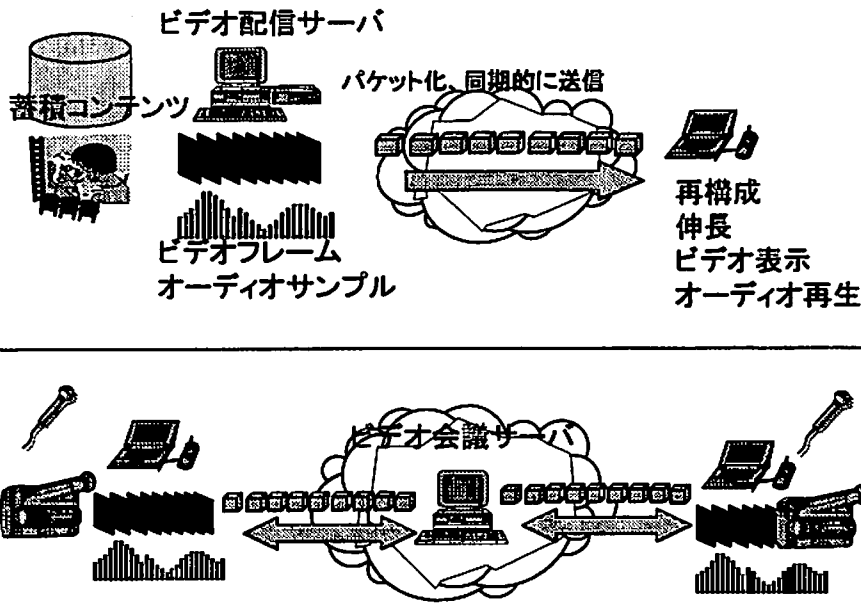


図 2.5: ビデオ、オーディオ通信

めのストレージの容量も必要である。モバイルユーザは情報をすぐに必要としているし、携帯端末はストレージ容量が限られているので、この手順には問題がある。

ビデオ、オーディオなどのマルチメディア情報は時系列的に変化する連続データであり、ビデオは一定間隔で連続するビデオフレーム、オーディオは一定間隔で連続するオーディオサンプルから構成される。ビデオ、オーディオを通信でやりとりする場合、送信者はビデオフレーム、オーディオサンプルをパケット化して、あるレートで同期的に送出する(図 2.5)。一般的にビデオフレームはいくつかの packets に分割されて配送され、オーディオサンプルはいくつかまとめて 1 つの packets に埋め込まれて配送される。また、ビデオ、オーディオ情報は容量が大きく、そのままでは高速なハードウェアとネットワークがなければ扱えないので、圧縮される。ビデオは空間的、時間的冗長度を削減する技術により圧縮される(それぞれフレーム内圧縮、フレーム間圧縮と呼ばれる)。オーディオは、人間の音響認知特性を利用して不必要な信号を削減し、圧縮される [43]。

受信者は packets を受け取り、元のビデオフレームやオーディオサンプルを再構成し、圧縮されている場合はその伸長を行う。そして、元々のビデオ、オーディオ情報を再現するために、元のタイミングでビデオフレーム、オーディオサンプルを再生する。このため、あるビデオフレーム、オーディオサンプルは受信者の再生タイミングまでに通信により届けられなければならない。その再生タイミングより遅れたり、落ちると、映像がぎくしゃくしたり、音声途切れたりしてしまう。

前述の理由から、ビデオ、オーディオのような実時間制約を持つ連続データの通信を行う場合、通信路は連続データが要求する通信サービス品質(データ転送速度、誤り率、遅延など)を保証しなければならない [44]。[45] では、ユーザが要求するメディア再生品質を、通

信路に要求する通信サービス品質にマッピングする手法について述べている。メディア再生品質として、画像サイズ、画質、滑らかさ、応答性などが設定され、それがスループット ([45]では平均パケットレートと読んでいる)、許容ジッタ、許容遅延、許容パケットロスといった通信サービス品質のパラメータに変換される。例えば、フレームサイズ 1920 × 1080(ピクセル)、フレームレート 30(fps) のビデオに対して、画像サイズをワイドに、そして高画質で非常に滑らか、かつ応答性を良く再生して欲しいというメディア再生品質要求は、スループット 4560(パケット/sec)、パケットサイズ 4096 (バイト)、許容パケットロス率 10^{-3} 、許容ジッタ 1/60(sec)、許容遅延 0.2(sec) といった通信サービス品質要求に変換される。

無線通信におけるマルチメディア通信

しかし、無線通信においては雑音、干渉、フェージングなどにより誤り率の増加、スループットの低下、遅延の増大といった通信サービス品質が低下しうる。また、低速であるという問題点がある。

そこで、誤りに強く、低ビットレートで、生成データ量を調節できるスケーラビリティを提供する圧縮符号化方式や、FEC,ARQ などの誤り制御を用いたビデオ転送などが検討されている [7][8][12][9]。また、ITU-T 勧告の低ビットレートのビデオ会議用映像符号化方式 H.263[10] や、ボイス符号化方式 G.729[46][47]、ISO/IEC JTC1/SC29/WG11 の MPEG-4(現在標準化作業中、最新情報は <http://www.cselt.stet.it/mpeg/> にある) などを利用できる。

インターネットにおけるマルチメディア通信

インターネットにおけるマルチメディア通信のために、RTP(RealTime Protocol)[24] が開発された。RTP はビデオ会議にも対応するためマルチキャストを意識して設計され、動作が軽く、信頼性を保証せず、コネクションレスであるという特徴を持ったトランスポートプロトコルである。マルチメディア情報の受信者が元のマルチメディア情報のタイミングを再現して再生を行うためにパケットにタイムスタンプを押すのが主な役割である。RTP はコネクションレスのトランスポートプロトコルである UDP の上に実装される。

また、RTP の制御プロトコル RTCP(RTP Control Protocol) はフロー制御を行うために受信側から送信側にトラフィック監視情報をフィードバックする機構を提供する。インターネットは皆が資源を共有しているので、輻輳制御、すなわちネットワークの混雑を回避するメカニズムが必要である。TCP はウィンドウフロー制御を持つが、UDP はそのような仕組みを持たないため、UDP の上位層はパケット消失やスループットの変化などをトリガーにして、レート制御を行う必要がある。

RTP,UDP を用いてビデオ、オーディオをパケット化して、インターネット上で配送する際、問題となるのはパケットロスとジッタ(遅延の変動)である。インターネットはベストエフォートという概念に基づいて基本的に品質保証の仕組みを持たないので、ルータでの

パケット処理時間の変動、処理待ちパケット数の増減、経路変更に伴う遅延の変動などにより、パケットロス率の増加、スループットの低下、遅延の変動、増大といった通信サービス品質が低下する。ビデオパケット、オーディオパケットが失われたり、遅れると元のマルチメディア情報のタイミングが再現されず、結果としてビデオ、オーディオの再生に障害を及ぼす。

これに対し、ネットワーク側で通信サービス品質を保証しようとするアプローチとマルチメディアアプリケーションが通信サービス品質の変動に適応しようとするアプローチがある。

ネットワーク側で通信サービス品質を保証すれば、マルチメディアアプリケーションを単純に設計することができ、高品質なマルチメディア再生を期待できるが、経路上のすべてのノードが通信サービスを保証する制御機能（CBQ[15]やWFQ[16]などのキューイング、RSVP(RFC2205)[14]などのプロトコル、許諾制御など）を持つ必要がある。

一方の適応的アプローチは、アプリケーションがマルチメディア情報の空間的、時間的解像度を調節して消費帯域幅を削減し、スループットの変動に追従したり、バッファリングや再生点の調節などを行って、パケットの転送遅延の変動を吸収するものである。例えば、ビデオの受信者が送信者に達成されたフレームレートをフィードバックすることで、送信側は送信フレームレートを調節することにより、送信ビットレートやCPU処理量を減らすことができる [19][20]。フィードバックは、アプリケーションレベルのプロトコルで行うこともできるし、前述の RTCP の機構を用いることもできる。

送信者はフレームレートだけでなく、色数を調節したり（例えばカラーからモノクロなど）、画像の空間的解像度を調節して送信ビットレートを増減することができる。IVS[21],vic[22]はH.261符号化方式を利用するインターネット上のビデオ会議システムで、動的に映像の出力データ量を制御し、スループットの変動に適応することができる。また、階層的符号化[23]の技術もネットワーク環境への適応に利用できる。UNIX環境の音声会議ツールvatはボイスデータのバッファリングや、再生点の調整によってボイスパケットの転送遅延の変動に適応する機能を持つ。

適応的手法はエンドエンドのホストに組み込めば利用可能であり、広範囲なインターネット、モバイルコンピューティング環境に適用可能である。ただし、マルチメディアアプリケーションの設計が複雑になり、適応能力にも限界がある。

圧縮符号化方式に関しては、無線通信と同様にインターネットでも、ITU-T勧告の低ビットレートのビデオ会議用映像符号化方式H.263[10]や、ボイス符号化方式G.729[46][47]、ISO/IEC JTC1/SC29/WG11のMPEG-4(現在標準化作業中、最新情報は<http://www.csel.tstet.it/mpeg/>にある)などが利用される。より高品質だが比較的高いビットレートを要求するMPEG1,MPEG2[43]も利用できる。他にも、QuickTimeやReal Video¹⁰などの圧縮符号化方式は、ほぼ業界標準としての地位を確立している。

¹⁰<http://www.real.com/>

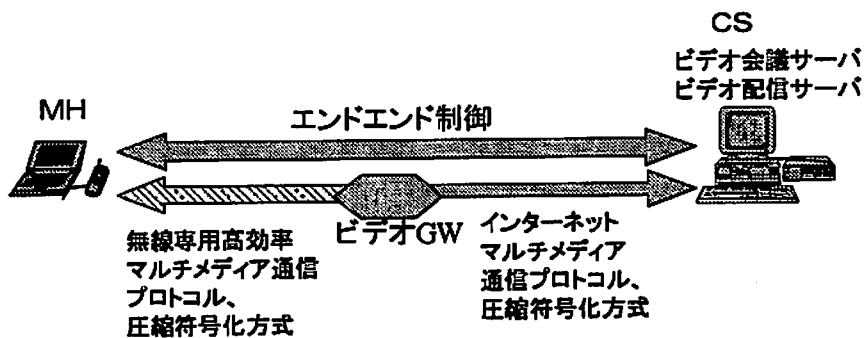


図 2.6: モバイルマルチメディア通信アーキテクチャ

モバイルマルチメディア通信アーキテクチャ

本論文は、広域無線通信とインターネットをまたがったネットワーク環境を考えている。2.4.3、2.4.3で述べたように無線、インターネット、それぞれにマルチメディア通信の技術が開発されている。そこで、インターネットの枠組みに基づくエンドエンド制御のビデオ転送と、無線とインターネットそれぞれの技術を利用するスプリットビデオ転送の2つのモバイルマルチメディア通信アーキテクチャが考えられる (図 2.6)。

- エンドエンド制御ビデオ転送: MH は CS と直接、ビデオやオーディオをやりとりする。無線通信部分もインターネットの一部とみなし、MH と CS 間でインターネットのマルチメディア通信プロトコル、圧縮符号化方式を利用して、エンドエンドでフロー制御やレート制御などを行う。
- スプリットビデオ転送: 無線部分では無線専用の高効率マルチメディア通信プロトコル、圧縮符号化方式を用い、インターネット上ではインターネットのマルチメディア通信プロトコル、圧縮符号化方式を用いる。イントラネット、ISP 内のユーザのインターネットへのアクセスポイント (RAS など) にビデオ圧縮符号化方式の変換処理、プロトコル変換処理を行うビデオ GW を配置する。

エンドエンド制御ビデオ転送の利点は、スプリットビデオ転送に比べて単純で、特別な設備 (ビデオ GW) を必要とせず、広範囲で利用可能であることである。欠点としては、無線部分で携帯電話の PDC 方式、PHS の PIAFS などそのまま利用するため、無線上のビデオ

オ通信に特化した高効率のマルチメディア通信プロトコル、圧縮符号化方式を利用するのに比べ、性能が落ちる可能性がある。

一方、スプリットビデオ転送の利点は、当然ながら無線部、有線インターネット部、それぞれに適したマルチメディア通信プロトコル、圧縮符号化方式を利用できる点である。欠点としてビデオ GW に関する様々な問題が考えられる。ビデオ GW の配備コストがかかるので広域での利用可能性に問題があること、ビデオ電話のようなライブビデオ通信の場合、ビデオ GW の変換処理による遅延が問題になるかもしれないこと、ビデオ GW はいくつかの MH にサービスする必要があり、そのスケラビリティに問題があるかもしれないこと、PHS、携帯電話など無線通信ごとに専用のビデオ GW が必要であることなどが挙げられる。

本論文は、特別な設備を必要とせず、広範囲で利用可能なエンドエンド制御ビデオ転送の通信アーキテクチャを採用する。データリンク層以下はある無線通信を仮定することなく、低速、不安定で高遅延、品質が変動するリンクであるとみなして、ネットワーク層 (IP 層) 以上のマルチメディア通信方式を議論する。2.4.3で述べたアプリケーションのレベルの適応的手法と、ネットワーク層における通信サービス品質の向上の手法の両方を検討する。

適応的手法として選択的マルチメディア通信方式と先読み方式を提案する。選択的マルチメディア通信方式は蓄積型のビデオコンテンツの意味的重要度に基づいてビデオフレームに優先度を与え、スループットが低下した場合に、優先度の低いフレームを落とすような選択的なビデオフレームの転送を行う。先読み通信方式は、蓄積型ビデオの取得の際、受信者に常に時間的に先のビデオ情報を与え、ストレージに保持させ、インターネット上や無線通信上で突然、スループットや遅延などの通信サービス品質が低下しても、ビデオ再生品質を保つ。

一方、通信サービス品質の向上のために、実時間指向優先転送方式を提案する。MH、CS を含む MH-CS 間の各ノードにおいて、ネットワーク層でパケットの優先度と実時間制約に基づいたキューイングとパケットスケジューリングを行うことで、制御情報の実時間通信、ビデオ、ボイスなどのマルチメディア通信に対し、ベストエフォートの実時間通信サービス品質を提供する。

なお、無線通信を含むインターネットで、2.4.3で述べたインターネットのマルチメディア通信プロトコル UDP、RTP を利用することに関して、信頼性の問題が懸念されるが、大きな問題はないと考えている。UDP は再送を行わないが、PDC や PIAFS などのデータ通信プロトコルは信頼性向上のため、データリンク層で ARQ を使って誤り回復を行う。そこで、無線部分の誤りは (再送により遅延は増加するものの) 隠蔽され、UDP でもある程度の信頼性が保証される。

プロトコル

2.4.3で述べたように、本論文はインターネットの通信プロトコルに基づく。ビデオ、オーディオ転送や制御メッセージのやりとりにはインターネットの標準プロトコルである IP (IPv4: RFC791 [37], IPv6: RFC1883[48]), UDP (RFC768), RTP (RFC1889)[24], TCP (RFC 793), IP Multicast (RFC 1112)[49] を用いる。インターネット標準を移動ホストをサポート

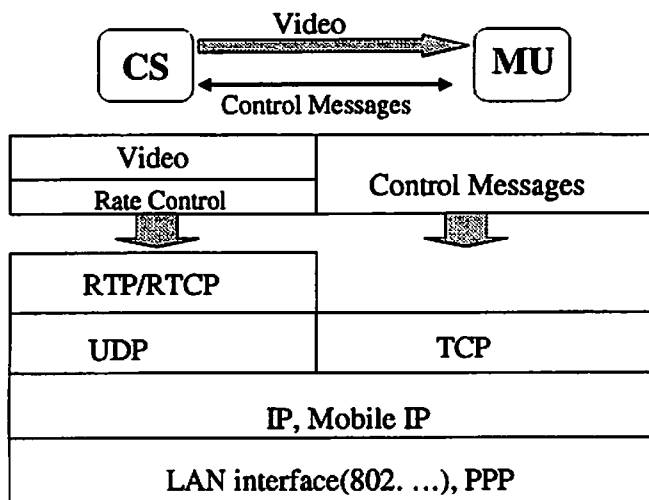


図 2.7: プロトコルスタック

トするためには IETF Mobile IP(RFC 2002)[28] や DHCP(RFC1541)[30] を利用することができる。図 2.7 にプロトコルスタックを示す。

2.5 あとがき

本章では、まず、モバイル通信アーキテクチャを紹介した。低速で不安定で、通信コストの高い無線通信におけるモバイルコンピューティングを実用的なものにするため、イントラネット内にエージェントを配置し、モバイルミドルウェアを携帯端末とエージェントに配備し、アプリケーションも非同期操作（切断時操作）に対応するものである。

次に、本論文が対象とするネットワーク環境を示した。モバイル通信アーキテクチャはイントラネットと広域無線通信の連携するネットワーク環境を想定しているが、本論文はイントラネットを含む広域インターネットと広域無線通信が連携するネットワーク環境を想定している。ユーザがインターネットを移動し、インターネットから頻繁に切断する。このモバイルコンピューティング環境において、マルチメディア通信を行うにあたり、移動性、切断性、通信サービス品質の 3 つの課題を解決する必要があることを述べた。

その内、移動性については DHCP, Mobile IP, 無線通信システムの移動サポートなどの既存の技術で解決できることを示した。次章で、切断性を解決するための同期・非同期通信方式、通信サービス品質の課題を解決する選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式という 4 つのマルチメディア通信方式を提案する。

第 3 章

マルチメディア通信方式

3.1 まえがき

本章は、2.4で述べた残る課題、すなわち切断性の課題、通信サービス品質を解決するマルチメディア通信方式を提案する。同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式という4つのマルチメディア通信方式を提案する。

それぞれのマルチメディア通信方式の扱う問題と、ライブ型（同期型）、蓄積型（非同期型）のどちらのマルチメディア通信形態に適用可能かについて図 3.1に示す。同期・非同期通信方式は、ビデオ会議システムやビデオ電話などにおけるヒューマンコミュニケーション（会議）が、ユーザの都合や通信の切断により中断される問題を改善する。選択的マルチメディア通信方式と先読み方式は低速で、通信サービス品質が変動するという問題を扱う。両方式は蓄積型のマルチメディア通信に適用可能である。実時間指向優先転送方式は、ライブ型と蓄積型の両方に適用可能であり、限られた通信容量を効率的に利用する仕組み、ベストエフォートの実時間通信サービスを提供する。

以下、3.2で同期・非同期通信方式、3.3で選択的マルチメディア通信方式と先読み方式、3.4で実時間指向優先転送方式について論じる。

3.2 同期・非同期通信方式

3.2.1 まえがき

コンピュータネットワークの発展と端末のマルチメディア化により、地理的に離れた相手と、ビデオ、ボイス、ホワイトボード、テキストなどを使ってコミュニケーションをとりながら協同執筆、ソフトウェア共同開発、電子会議などの協同作業を行ったり、個人的なコミュニケーションをとることが可能になった [50]。

そして近年、携帯電話、PHS など無線通信インフラの発展により、必要なときに所望の人と接続したり、情報を発信、受信することのできるモバイルコンピューティング環境が

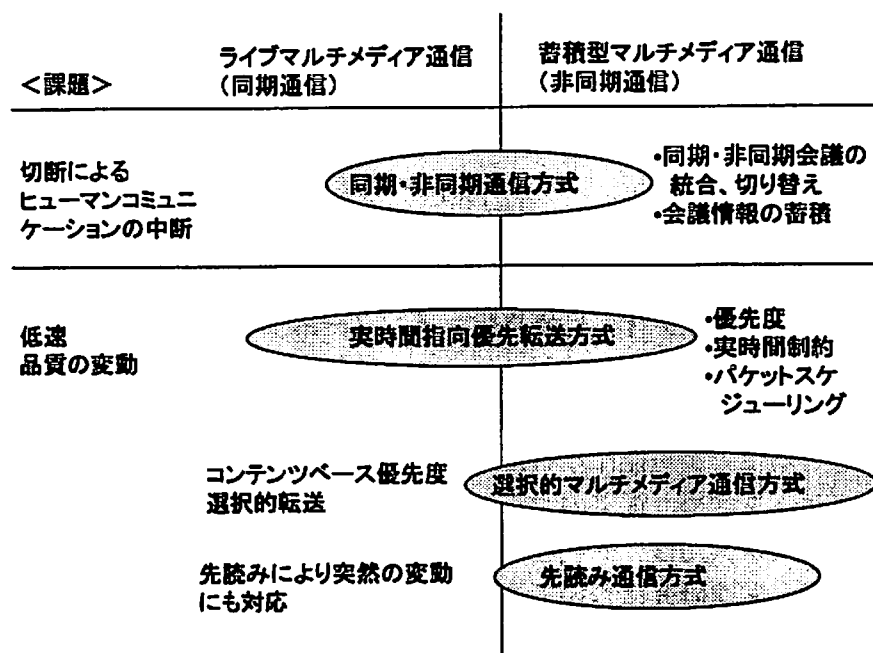


図 3.1: マルチメディア通信方式

整備されつつある [2]. PHS や携帯電話のデータ通信機能を使って、電車内などの移動中や出先の喫茶店などからインターネット,あるいはイントラネット上のマルチメディア電子会議システムに接続し,携帯端末を使ってミーティングや協同作業に参加したり,個人的なコミュニケーションをとることができる。

しかし,モバイルコンピューティング環境ではお互いが時間を共有するリアルタイムの協同作業,個人的コミュニケーションは,しばしばユーザの都合や通信路の状態の悪化により中断され,円滑なヒューマンコミュニケーションが阻害される。ユーザが電車を乗り換えたり,歩いて移動しているときには協同作業には参加できないし,高速移動中や無線通信のサービスエリア外にいるために無線通信の接続が切断しては通信することができない。

そこで,ヒューマンコミュニケーションを伴うマルチメディア通信では,ユーザの都合や通信路の状態に起因するコミュニケーションの中断を前提として,円滑なコミュニケーションを達成するためのサポートを検討する必要がある。

本論文では,協同作業や個人的コミュニケーションなどのヒューマンコミュニケーションを会議と呼び,モバイルコンピューティング環境における会議の実用性,使い勝手を向上させるために,同期・非同期通信方式を提案する。同期・非同期通信方式は,1) 会議形態を同期,非同期に切り替えることのできる自由度,2) 部分非同期会議の導入により同期会議から一時的に欠席することを許す柔軟性,3) ホームサーバの導入により,ネットワークに接続していないユーザへ送信することを可能にする利便性,を提供する。また,同期・非同期通信方式は,やりとりされる情報が協同作業における共有情報か,ビデオ電話のような個人的情報かにより,集中型と分散型の2種類の蓄積方法を使い分ける。

同期・非同期通信方式は,ビデオ会議システム,ビデオ電話システムのクライアントと

サーバに組み込まれるモジュールと、各ユーザのホームネットワーク内のホスト上で実行されるホームサーバと呼ばれるアプリケーションにより実現される。

3.2.2 背景

以前のビデオ会議システムは、高価なワークステーション、ビデオカメラを必要とし、ISDN、LAN などの高品質なネットワークを利用することを想定して設計されていた [50]。しかし、近年、パソコンのマルチメディア化、低価格化、インターネットの普及により、CU-SeeMe、NetMeeting などの安価なインターネットパソコン会議システム [51] が登場し、個人ユーザもマルチメディア電子会議システムを利用して協同作業を行ったり、ビデオ電話としてコミュニケーションをとることが可能になった。

しかし、パソコン会議システムも含め、従来のマルチメディア電子会議システムは、モバイルコンピューティング環境では実用性、使い勝手に問題がある。以下で、本論文の前提条件、定義を述べた後に、課題、関連研究について述べる。

前提条件、定義

本論文では、協同作業や個人的コミュニケーションなどのヒューマンコミュニケーションを会議と呼ぶ。本論文で扱う協同作業、個人的コミュニケーションは、同僚や仲間、友人との打ち合わせ、伝達、ブレインストーミングなどのインフォーマル、カジュアルなミーティングであり、時間的に融通がきき、通信が切断してしまったり、少しの間席を外すなどの多少の障害が許容されるものとする。会議参加者数は2～4人までの少数を対象する。少しの障害も許されないようなフォーマルな会議、多人数の会議は、モバイルコンピューティング環境、無線通信環境で実施するのは困難であると考ええる。

本論文では、全参加者が会議中、継続して会議システムに接続し、全参加者が同じ時間を共有してリアルタイムに発言をやりとりする会議を同期会議と呼び、その参加者の参加形態を同期参加と呼ぶ。一方、参加者が都合のよい時間に関欠的に会議システムに接続し、非同期に発言をやりとりする会議を非同期会議と呼び、その参加者の参加形態を非同期参加と呼ぶ。非同期会議では会議システムのストレージに発言を蓄積することで、同じ時間を共有しない参加者が発言をやりとりすることを可能にする。

さらに、本論文では同期参加する参加者と非同期参加する参加者が混在する部分非同期会議形態を導入する [52]。非同期参加する参加者は不在期間の会議進行に関し、他の参加者に説明を求めることができるが、その詳細まで知るのは難しい。そこで部分非同期会議においては、同期参加、非同期参加に関係なく、すべての参加者の発言をストレージに記録する。非同期参加する参加者は蓄積情報を参照して会議進行の詳細を知ることができる。

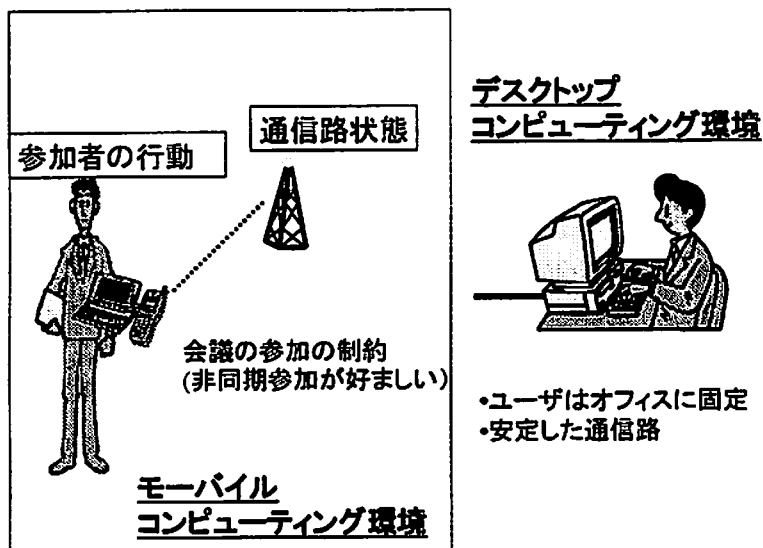


図 3.2: モバイルコンピューティング環境と固定環境

課題

モバイルコンピューティング環境にいる参加者は、その参加者の行動、通信路状態により会議に参加できなくなるという制約を受ける(図 3.2)。画面を見たり、ホワイトボードやテキストの書き込みを行うには、座って携帯端末を操作する余裕が必要であり、電車を乗り換えたり、歩いて移動しているときなどは会議に参加することができない。無線通信が不安定で接続が切れてしまったり、高速移動中や無線通信のサービスエリア外にいて、通信できない場合も会議に参加することができない。移動中、出先にいるユーザは、同期会議よりも自分の都合の良いときに参加できる非同期会議の方が望ましい。

そこで、モバイルコンピューティング環境における会議には、ユーザが同期参加するのが難しくなったら非同期会議に切り替えて会議を続行し、その後、同期参加が可能になったら同期会議に切り替える自由度が必要である。もちろん、会議形態の切り替えは参加者同士の同意が必要であり、特に非同期会議から同期会議の切り替えは呼び出された相手が不在であったり、拒否するかもしれない。しかし、会議形態を切り替える仕組みは有用であると思われる。また、一時的に会議から欠席しても会議進行に支障をきたさないように、部分非同期会議を導入することも必要である。会議情報が蓄積されていれば、ある参加者が一時的に会議から離れても、あとから不在期間の会議進行を知ることができる。

しかし、これまで同期会議、非同期会議の両方をサポートするマルチメディア電子会議システムはほとんどなかった。同期会議をサポートする会議システムと非同期会議をサポートするシステムを使い分けるのは、ユーザに負担をかけるかもしれないし、ホワイトボードなどの共有空間を別システムで引き継ぐことができないかもしれない。非同期会議をサポートする電子会議システムで準リアルタイムの同期会議を行うのも使い勝手に難があると思われる。

協同作業ではないが、同期と非同期のヒューマンコミュニケーションを統合したシステム例として、Talkware[53]がある。これは電子メールと電話を統合したLANベースのシステムで、電話をかけて相手が不在の際は、発信者はそのままの状態でもメールを起動してボイスやテキストでメッセージを送ることができる。携帯電話、PHSのサービスで、電話以外に電子メールを含む文字メッセージやりとり機能を提供するものがある。筆者らもこれまで、同期と非同期両方の会議をサポートする同期・非同期電子会議システムPARCAE[52]を開発してきた。本論文で提案する同期・非同期通信方式は、PARCAEの概念に基づくもので、ビデオ会議システム、ビデオ電話などのヒューマンコミュニケーションを伴うマルチメディア通信システムに組み込まれるモジュールとして実現され、同期会議、非同期会議、部分非同期会議の3つの会議形態の切り替えや、会議情報の蓄積機能を提供する。

3.2.3 同期・非同期通信方式

同期・非同期通信方式は、ユーザの都合や通信路の状態によりコミュニケーションが中断される問題を改善するという問題に対して以下のような機能を提供する。

- 会議形態の切り替え：全参加者が同じ時間を共有する同期会議と、各参加者が都合の良いときに参加する非同期会議を必要に応じて切り替える。
- 会議情報の蓄積：ユーザが同期会議から一時的に欠席することを許す部分非同期会議を導入する。参加者は一時的に会議から離れても、後から蓄積された情報を取得し、不在期間の会議進行を知ることができる。
- 切断中のユーザへの送信：各ユーザのホームネットワーク上にユーザを代理するホームサーバを配置する。ホームサーバは常時インターネット上に存在し、ユーザ端末がネットワークから切断しているときは、留守番電話のように代わりに受信、蓄積を行う。

システム構成

同期・非同期通信方式は、ビデオ会議システム、ビデオ電話システムのクライアントとサーバに組み込まれるモジュールと、各ユーザのホームネットワークのホスト上で実行されるホームサーバと呼ぶアプリケーションにより実現される。ビデオ会議システム、ビデオ電話システムには、会議サーバを介してクライアント(ユーザ端末)が通信するシステム構成と、会議サーバなしにクライアントが対等に通信を行うシステム構成がある。同期・非同期通信方式は、非同期会議、部分非同期会議をサポートするために、会議情報の蓄積を行うサーバを必要とする。

同期・非同期通信方式は図3.3のように、クライアントと、共有の会議サーバ、各ユーザ専用のホームサーバという2つのサーバを含むシステム構成をとる。会議情報をどこに蓄積すべきかは、やりとりされる情報が共有のものか、それとも個人的なものかにより異なると考え、共有情報を蓄積する共有の会議サーバと個人的情報を蓄積するホームサーバをシステ

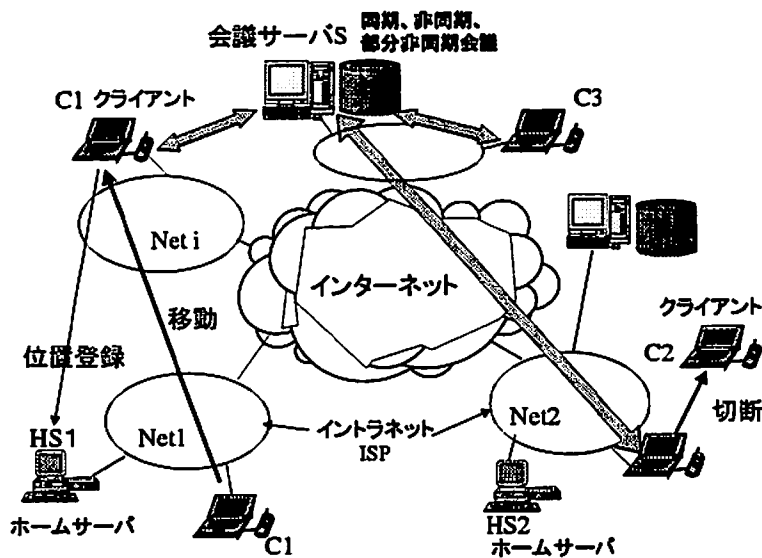


図 3.3: システム構成

ムに組み入れた。同期・非同期通信方式は、ユーザ自身が会議の性質を判断して、会議開始時に集中型の蓄積か、分散型の蓄積かを指定することを仮定している。

協同作業のように共有情報を伴う通信の場合、共有の会議サーバに蓄積する集中型の蓄積が適している。各自が蓄積するのに比べ、ストレージを有効利用することができるし、誰かがホワイトボードなどの共有情報を更新しても常に最新情報を得ることができる。一方、ビデオ電話のような個人的情報を伴う通信においては、情報を各ユーザのホームサーバに蓄積するような分散型の蓄積が適している。ユーザは蓄積された情報を個人的に所有し、自由に加工、削除することができる。

以下で、同期・非同期通信方式を実現する、クライアントと会議サーバに組み込まれるモジュール、ホームサーバと呼ばれるアプリケーションの概要を述べる。

クライアント クライアントはユーザ端末上で動作するアプリケーションで、マルチメディア情報の入出力を行う。ビデオカメラ、マイクからビデオ、ボイスデータを入力し、サーバに対して送信する。逆に、サーバから他の参加者のビデオ、ボイスデータを受信し、表示、再生を行う。同様に、マウス、ペン、キーボードによるホワイトボード、テキストの書き込みを入力し、サーバに送信したり、受信して表示する。

同期・非同期通信方式は、各クライアントにユーザの参加形態要求を受け付ける参加者エンティティ PE (Participant Entity) というモジュールと、通信路ウォッチャ LW (Link Watcher) というモジュールを組み込む (図 3.4)。LW は各クライアントと会議サーバの組の両端に配置され、通信路の状態を監視し、対応するユーザ端末が切断していないかを調べるものである。クライアント側の LW を LWc、サーバ側の LW を LWs と記す。

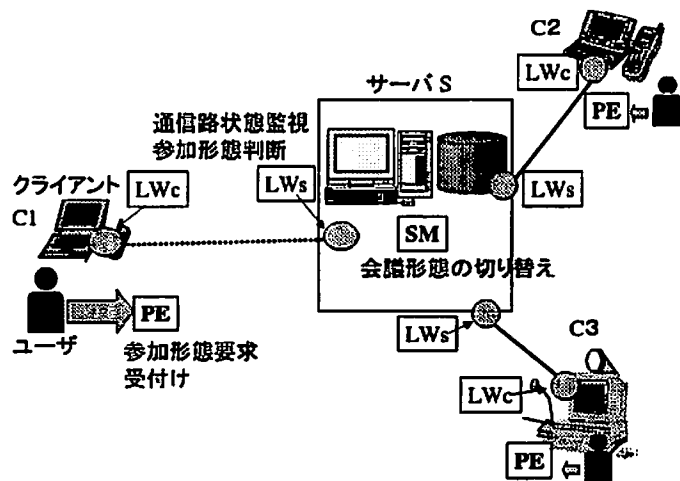


図 3.4: モジュールの組み込み

会議サーバ 会議サーバはインターネット上、イントラネット上に配置されるアプリケーションで、ユーザの参加（退席）状況の管理等の会議制御、マルチメディアデータ（ビデオ、ボイス、ホワイトボード、テキストデータ）の多地点配送を行う。

同期・非同期通信方式は会議サーバにセッションマネージャ SM(Session Manager) というモジュールと、会議サーバ側の通信路ウォッチャ LWs というモジュールを組み込む(図 3.4)。SM は PE から通知される各ユーザの参加形態要求と、LWs から通知される各ユーザの接続・切断状態情報から、会議形態の決定、切り替えを行う。会議形態が非同期、部分非同期で、ユーザが集中型の蓄積を指定していた場合、SM が会議情報の蓄積を行う。ビデオ、ボイス、ホワイトボード、テキストの書き込みをストレージに蓄積し、あるユーザが会議サーバに蓄積情報の取得要求を行ったら、配送する。一方、分散型の蓄積が指定されていた場合、SM は非同期参加しているユーザのホームサーバに蓄積を命じる。

ホームサーバ 各ユーザは自分の組織（学校、会社）、あるいは契約している ISP のホームネットワークのホストにホームサーバ HS と呼ばれるアプリケーションを配置する。HS は常時インターネット上に存在し、ユーザのアクセスポイントとしての役割を果たす。HS がクライアント C を呼び出すことができるように、クライアントはホームネットワーク以外に接続（移動）したとき、常に現在の IP アドレスを自分の HS に伝えるとする。

ビデオ電話のような個人的コミュニケーションを行っていて、分散型の蓄積が指定された場合、非同期参加しているユーザの HS は SM から蓄積を命じられる。ユーザはネットワークから切断していたり、同期参加する余裕がない場合、会議の参加形態は非同期参加となる。HS はユーザの代わりに会議サーバに接続し、留守番電話のように受信、蓄積を行う。

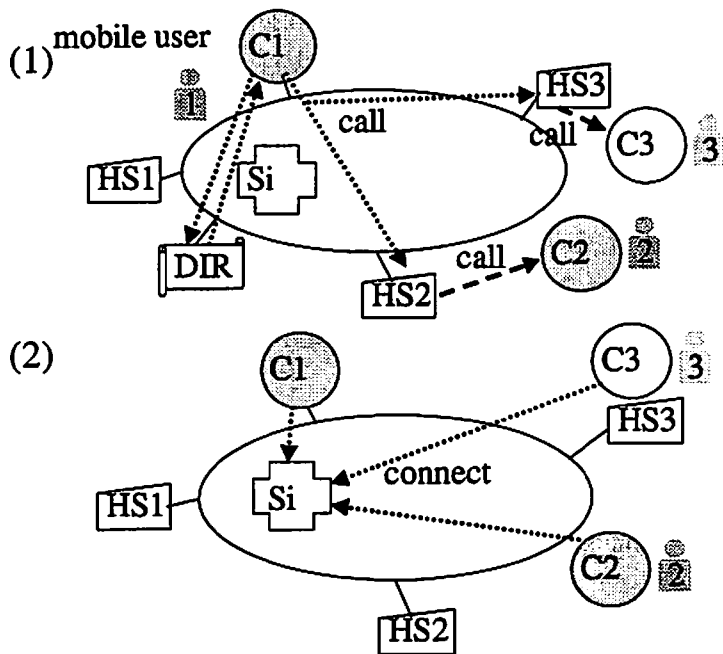


図 3.5: 会議の開始手順

会議の開始手順

会議の開始手順を図 3.5に示す。まず、仮定として、インターネット上にいくつかの会議サーバ S が配置されているとする。また、インターネット上にユーザ名と HS を結びつけるようなディレクトリサービス DIR(Directory Service) が存在すると仮定する。本論文では DIR の詳細については扱わない。例えば、Microsoft NetMeeting¹では、いくつかのホスト上でディレクトリサーバが提供されている。ディレクトリサーバにアクセスすると、現在、通話可能なユーザの一覧が、ユーザの名前、メールアドレス、住んでいる地域と共に表示される。

ユーザ 1 がユーザ 2,3 と会議サーバ S を利用して、集中型の蓄積を行う同期会議を始めようとしたとする。ユーザ 1 はユーザ 2,3 の HS のアドレスを知っている場合、会議サーバ S を指定して、呼び出す。アドレスを知らない場合は DIR に問い合わせ、HS2,HS3 のアドレスを取得した後、呼び出しを行う(図 3.5(1))。HS2, HS3 はユーザ 1 から会議サーバ S で会議を開くという呼び出しがあったことを、ユーザ 2,3 (クライアント C2, C3) に通知する。ユーザ 2 はホームネットワークから移動しているが、HS2 はユーザ 2 がどこにいるかを把握している(C2 は移動先から HS2 に自分の居場所を通知済みである)。

次にユーザ 1,2,3 が S に接続し(図 3.5(2))、同期会議が始まる。なお、利用するサーバ S を選ぶ際は、参加者全員から見て公平な、あるいはもっとも近いネットワーク上の位置にあるサーバを選ぶのが望ましいが、その探索は本論文の対象外である。

以上は、ユーザ 1 がユーザ 2,3 を呼び出したとき、ユーザ 2,3 がネットワークに接続して

¹<http://www.microsoft.com/netmeeting/>

いて、かつ呼び出しを受け入れた場合であり、すべてのユーザが同期参加している。しかし、実際には、ユーザはネットワークから切断中であるかもしれないし、都合により呼び出しを拒否するかもしれない。この場合、そのユーザは会議に非同期参加することになり、部分非同期、または非同期の会議形態で会議が始められる。また、同期会議が始まった後、部分非同期、または、非同期会議に切り替わることもある。3.2.4で、会議開始時の会議形態の決定、会議途中の会議形態の切り替えの仕組みについて述べる。

3.2.5では、非同期会議、部分非同期会議時の会議情報の蓄積について述べる。同期・非同期通信方式は、扱う情報によって集中型と分散型の2種類の蓄積方法を使い分ける。

3.2.4 会議形態の切り替え

セッションマネージャ SM は、会議開始時の会議形態を、各クライアントの参加者エンティティ PE から通知される各ユーザの参加形態要求から決定する。会議開始後は、各ユーザの参加形態要求と、通信路ウォッチャ LW から通知される各ユーザの接続・切断情報から、会議形態を決定する。

参加者エンティティ PE

各ユーザは他のユーザから会議への呼び出しを受けたとき、あるいは現在の自分の参加形態を変更したいとき、自端末のクライアントの参加者エンティティ PE に同期参加するか、非同期参加するかという参加形態要求を PE に通知する。PE はクライアントに組み込まれるモジュールであり、ユーザはクライアントアプリケーションのボタンを押すなどして、参加形態要求を PE に通知する（ユーザインタフェースは実装に依存する）。PE はユーザの入力を受け取ったら、すぐに SM に通知する。

通信路ウォッチャ LW

通信路ウォッチャ LW は同期参加を要求しているユーザの接続状態を監視し、SM に通知する。クライアント上の LWc は、同じクライアントに存在する PE の参加形態要求を参照し、そのユーザが同期参加を要求している場合、サーバ上の LWs に対し、一定間隔（例えば5秒ごと）でメッセージを送る。LWs は LWc からメッセージを受け取ったら、そのユーザは接続していると判断する。しかし、前回メッセージを受け取ってからタイムアウト時間（例えば30秒）を越えてもメッセージが到着しない場合、切断していると判断する。LWs はクライアントの接続状態が（接続か切断か）変化したとき、SM に通知する。

セッションマネージャ SM

本論文では一連の継続する会議をセッションと呼ぶ。あるユーザが他のユーザを呼び出して会議を開催するとセッションが開始され、ユーザが会議を完全に終了するときにセッションが削除される。会議形態が同期、非同期、部分非同期に切り替わっても、セッションは維

	PE: 同期参加要求	PE: 非同期参加要求
LW: 接続	同期参加	非同期参加
LW: 切断	非同期参加	非同期参加

表 3.1: ユーザの参加形態の決定

持される。協同作業の場合、セッションのライフタイムは数日、数週間など比較的長くなるだろう。セッション中、会議参加者の全体リスト、蓄積された会議情報などが維持される。

会議開始時 セッションマネージャ SM は会議開始時に、各クライアントの PE から通知される各参加者の参加形態要求を受け取り、以下のアルゴリズムに従って会議形態を決定する。全員が同期参加を要求していれば同期会議、同期参加のユーザが 1 人以下ならば非同期会議、2 人以上ならば部分非同期会議に決定される。 m は全参加者数、 n は同期参加を要求しているユーザ数である。

$$\text{会議形態} = \begin{cases} \text{同期会議} & n = m \\ \text{部分非同期会議} & 1 < n < m \\ \text{非同期会議} & n \leq 1 \end{cases}$$

会議中 SM は会議中、SM は各ユーザの PE から参加形態要求を、LW から接続、切断情報を受け取ったら、各ユーザの参加形態を決定し直した後、会議形態を決定する。あるユーザの参加形態は、参加形態要求が同期参加であり、接続状態ならば同期参加である。一方、参加形態要求が非同期参加であるか、切断状態である場合、非同期参加と決定される(表 3.1)。次に、会議形態が会議開始時と同様に決定される。全員が同期参加ならば同期会議、同期参加のユーザが 1 人以下ならば非同期会議、2 人以上ならば部分非同期会議に決定される。

会議形態の切り替え例 ユーザ 1,2,3 で同期会議を行っている際、SM がユーザ 2 の PE から非同期参加という通知を受け取ったり、ユーザ 2 の LW から切断状態という通知を受け取ると、SM は上記のアルゴリズムにより、ユーザ 2 の参加形態を非同期参加にし、会議形態を部分非同期に切り替える。ユーザ 1,3 はユーザ 2 がいなくて同期会議をする意味がないと判断したら、非同期参加要求を自分の PE に通知する。同期参加するユーザが 1 人以下になったら SM は会議形態を非同期会議に切り替える。

その後、ユーザ 2 が会議に再参加したとする。ユーザ 2 は SM から蓄積された情報を取得し、不在期間の会議進行を知ることができる。部分非同期会議が続行されていた場合、ユーザ 2 が PE に同期参加要求すると、全員が同期参加になり、SM は会議形態を同期会議に切り替える。一方、ユーザ 2 が不在のために非同期会議に切り替えられていた場合、ユーザ 2 はユーザ 1,3 を呼び出して同期会議を再開することもできる。ユーザ 1,3 のどちらかが呼び出しを受理して同期参加すれば、会議形態が部分非同期会議に切り替わり、両方が受理して同期参加すれば、同期会議に切り替わる。

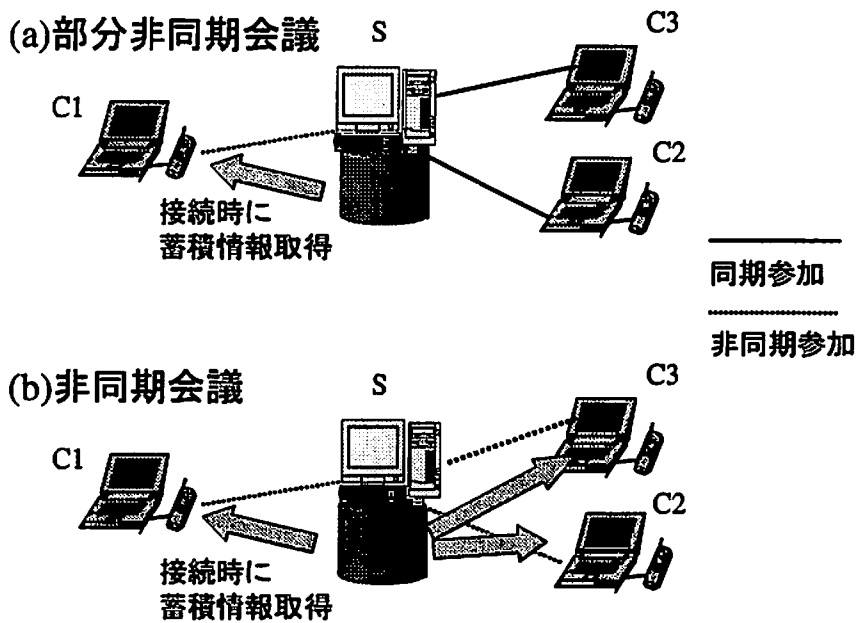


図 3.6: 集中型の蓄積

3.2.5 会議情報の蓄積

SM が会議形態を非同期会議、部分非同期会議に決定したとき、会議情報の蓄積が行われる。同期・非同期通信方式は、集中型と分散型の2種類の蓄積を行う。会議形態が同期会議のときは会議情報の蓄積は行われない。

集中型の蓄積

ユーザが集中型の蓄積を指定した場合、SM 自身が会議情報の蓄積を行う。SM は会議サーバがクライアントから受け取るすべてのマルチメディアデータ（ビデオ、ボイス、ホワイトボード、テキストの書き込み）をストレージに蓄積する。ユーザは一時的に会議から離れても、後から会議サーバに接続し、SM から蓄積された会議情報を取得して、不在期間の会議進行を知ることができる。

集中型の蓄積では、図 3.6のように会議サーバに会議情報が蓄積される。非同期参加しているユーザは、会議サーバに接続したとき、蓄積された会議情報を取得する。(a) 部分非同期会議ではユーザ1 (C1) が、(b) 非同期会議では全ユーザ (C1,C2,C3) が非同期参加している。

分散型の蓄積

一方、ユーザが分散型の蓄積を指定した場合、SM はすべての非同期参加しているユーザのホームサーバに蓄積を命じる。同期参加しているユーザのホームサーバは蓄積を行う必要

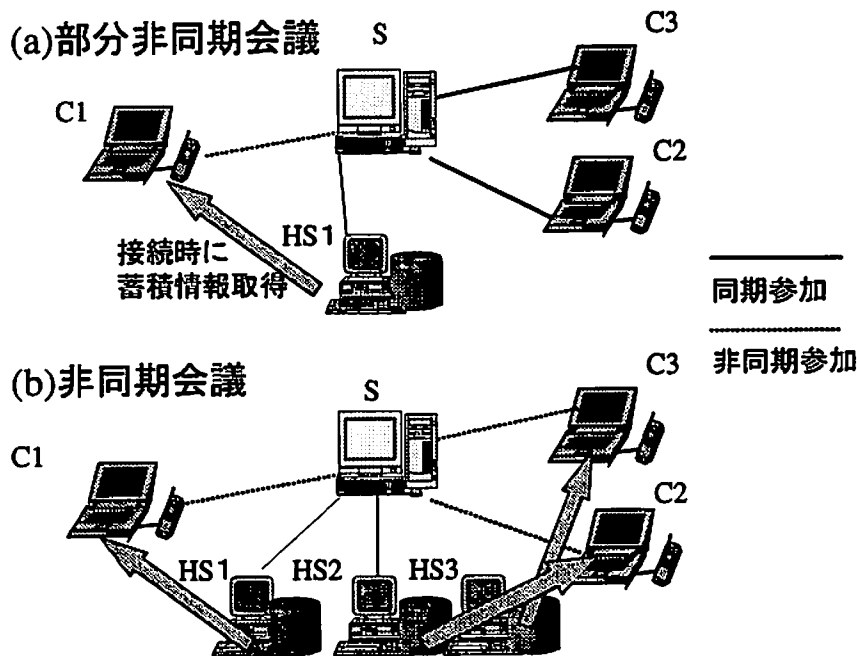


図 3.7: 分散型の蓄積

はない。図 3.7 のように非同期参加しているユーザの HS は会議サーバに接続して、ユーザの代わりに受信、蓄積を行う。同期参加しているユーザと同様にすべてのマルチメディアデータ（ビデオ、ボイス、ホワイトボード、テキストの書き込み）を受信し、ストレージに蓄積する。非同期参加しているユーザは自分の HS にアクセスして、その蓄積された情報を取得できる。

分散型の蓄積により、切断中の端末へメッセージを残すことができる。呼び出し時にネットワークから切断していたり、呼び出しを拒否したユーザは非同期参加の形態で会議に参加することになり、そのユーザの HS が代理で会議サーバに接続している。すべての発言は HS に蓄積され、非同期参加しているユーザが HS にアクセスしたとき、見聞きされる。

3.2.6 実装と評価

ビデオ会議システムのクライアントと会議サーバのプロトタイプを Microsoft Video for Windows 開発キットを利用して Windows 95 上に実装した。クライアント、サーバに同期・非同期通信方式の参加者エンティティ PE、セッションマネージャ SM に対応するモジュールを組み込んだ。クライアントと会議サーバは C++ 言語により記述され、クライアントのステップ数は 8210、サーバは 4952 である。プロトタイプは、ビデオ、ボイス、ホワイトボード、テキストによる同期会議、非同期会議、部分非同期会議を行うことができる。ホームサーバの実装に関しては今後の課題とし、本論文では扱わない。

クライアントアプリケーションは主にマルチメディア情報の入出力を行う。ビデオカメラ、マイクからビデオ、ボイスデータを入力し、サーバに対して送信する。逆に、サーバから

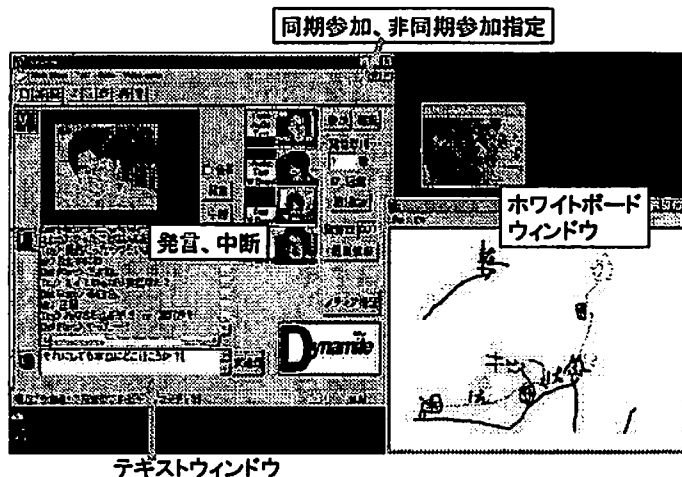


図 3.8: クライアント画面

他の参加者のビデオ、ボイスデータを受信し、表示、再生を行う。同様に、マウス、ペン、キーボードによるホワイトボード、テキストの書き込みを入力し、サーバに送信したり、受信して表示する。クライアント、サーバ間の通信は、ビデオ、ボイスに関してはUDP、ホワイトボード、テキスト、コンポーネント間の通知などに関してはTCPを用いている。ビデオはビデオフレームをいくつかのパケットに分割してUDPで配送し、ボイスは一つのパケットにいくつかのオーディオサンプルを入れてUDPで配送している。図3.8にクライアントのユーザインタフェースを示す。ユーザは参加、離脱ボタンにより、会議への同期参加、非同期参加をPEに通知する。

会議サーバは、ユーザの参加（退席）状況の管理等の会議制御とマルチメディアデータ（ビデオ、ボイス、ホワイトボード、テキストデータ）の多地点配送を行っている。多地点配送はIP Multicastなどのマルチキャストを利用せず、単純にユニキャストを複数回行うことで実現されている。

現在のプロトタイプは160×120ピクセル、4ビットグレースケール、2フレーム/秒、あるいは1フレーム/秒のビデオと、サンプリング周波数8KHz、量子化ビット8bitのボイスを扱うことができる。現在、圧縮を行っていないため、2フレーム/秒、1フレーム/秒のそれぞれのビデオの要求ビットレートは150Kbps、75Kbpsであり、ボイスは64Kbpsである。より低ビットレートのボイス、ビデオ圧縮符号化方式を利用すれば、要求ビットレートを下げることができる。

さらに、非同期参加するユーザが蓄積情報を取得する際のインデックスとして利用する蓄積会議情報のグラフ表示機能の実装を行った。クライアントは縦軸に参加者、横軸に時間をとって会議発言記録をグラフにプロットする(図3.9)。会議発言記録とは、サーバが各クライアントからビデオ、ボイス、ホワイトボード、テキストデータを受け取る際に記録する発言者、発言開始時刻、終了時刻、メディアの種類である。ビデオ、ボイスの発言は開始時刻、終了時刻に基づきバーで表示される。このグラフにより、誰がいつどれだけしゃべった

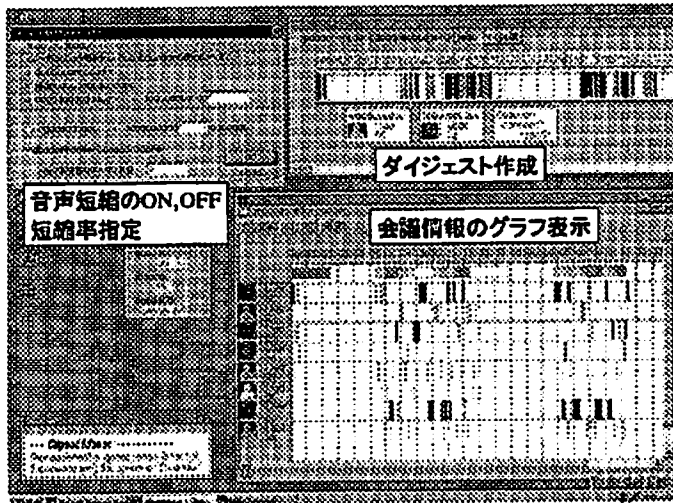


図 3.9: 蓄積会議情報のグラフ表示

かを視覚的に理解できる。ユーザがビデオ、ボイスを表すバーをマウスで選択すると、クライアントはサーバから対応するビデオ、ボイスデータを取得し、再生する。時間軸をマウスでクリックするとその時刻の近辺に発言されたテキスト、ホワイトボードの書き込みを見ることができる。

実験と考察

会議情報の蓄積 部分非同期会議、非同期会議の蓄積機能の有効性を確かめるため、開発したサーバ、クライアントを用いて会議実験を行った。会議サーバ、クライアントは研究室内の無線 LAN を含むイーサネット LAN 上に配置した。ボイスを利用して会議を行うため、広域無線通信により会議サーバにアクセスするクライアントを用いなかった。現在のプロトタイプボイスの要求ビットレートは、PHS や携帯電話の提供するデータ転送速度を越えているため、利用することができない。

会議の参加者は研究室の学生 5 名で、夏合宿の行き先、行き方についてボイス、ホワイトボード、テキストの 3 種類のメディアを用いて会議を行った (ビデオは利用しなかった)。会議形態は、4 名が同期参加し、1 名が非同期参加する部分非同期会議である。会議時間は 42 分で、ボイスの発言数は全部で 81 (平均約 12s の発言)、ホワイトボードの書き込み、テキストの書き込みは約 100 回あった。部分非同期会議のため、会議情報の蓄積が行われた。ボイスデータは 7.87MB、ホワイトボードは 12KB、テキストは 1KB のストレージ容量を消費した。現在のハードディスクの容量は数 GB が一般的であり、蓄積量は問題にならないと考える。

非同期参加するユーザは、会議サーバから蓄積されたボイス、ホワイトボード、テキストなどの会議情報を取得し、見聞きすることで、不在期間の会議進行を知ることができ、会議情報の蓄積機能の有効性が確かめられた。また、3.9 に示す会議蓄積情報のグラフ表示を見て、各参加者がどの程度発言したのかがよく分かった。参加者 4 の発言数が多く、会議の中

心となっているのが分かる。ただし、1つ1つのボイス発言を聞くのに時間がかかることが分かった。ホワイトボードやテキストなどは一覧性にすぐれ、比較的短時間で見ることができるが、ボイスは再生を聞かなければならない。今回の会議の場合、42分の会議中に発言されたボイス発言の時間の合計は約17分であり、すべてを聞くにはその時間かかる。

この問題に対し、[54]で蓄積された情報を効率的に取得するためのフィルタリングと音声短縮機能について提案した。フィルタリングはユーザが時間範囲、発言者、発言対象者の3つをキーとして、今から10分前までの発言、ある参加者Aの発言、自分に対してなされた発言の一覧が欲しいといった要求をすることを可能にする。音声短縮機能は、会議サーバ側で蓄積されたボイスデータのボイスサンプルを間引きしながら重ね合わせることでピッチを変えずにボイス発言の時間を短縮[55]し、蓄積されたボイス発言の取得データ量の削減とボイス発言を聞く時間の削減をはかるものである。

切断の検出 クライアントの接続、切断を判断する通信路ウォッチャの切断検出方法の実用性を確認するために、実験を行った。研究室のイーサネットLAN上に会議サーバを配置し、無線LANを介して会議サーバにアクセスするクライアント（無線LANクライアントと呼ぶ）、PHSの無線モデム通信によりリモートアクセスサーバを経由してサーバにアクセスするクライアント（PHSクライアントと呼ぶ）、それぞれについて10回の切断実験を行った。

クライアント上のLWcがサーバ上のLWsに対し、TCPを使って5秒間隔でメッセージを送る。タイムアウト時間を30秒と設定し、28Kbpsでビデオ転送を行っている途中で接続を切断した。無線LAN、有線LANクライアントについては物理的な切断を行い（無線LANのPCカードを抜いた）、PHSクライアントについてはソフトウェア的な切断を行った（Windowsのダイアルアップネットワークの切断を実行した）。

無線LANクライアントの場合、サーバ上のLWが受け取ったメッセージの間隔は平均5.67秒、最小値5.60秒、最大値5.83秒とほとんど一定であった。サーバが最後にメッセージを受け取ってから平均32.7秒後に切断を検出した。5秒よりも間隔が大きいのは、クライアントのタイマの割り込みとサーバの受信処理が、負荷が大きいと待たされるからであると思われる。現在の実装では、タイマの割り込みやデータの受信は、Windowsイベントとしてアプリケーションに通知され、負荷が大きいとイベントの通知が遅れる。高精度のタイマを利用すればこの状況を改善することができるが、切断の検出の目的においては現在の実装でも実用上問題はないと思われる。

一方、PHSクライアントは平均7.17秒、最小2.59秒、最大22.34秒と間隔が一定ではなかった。そして、平均33.1秒後に切断を検出した。PHSクライアントのメッセージの受信間隔に大きなばらつきがあるのは、PHSの転送能力を越えたビデオデータが送り込まれており、転送が滞っているためだと思われる。ビデオ転送のレートを半分の14Kbpsにしたところ、平均5.6秒、最小値5.49秒、最大値5.67秒とばらつきがなくなった。このように転送能力を越えたデータが流れていると、受信間隔のばらつきが大きくなる。タイムアウトを小さくすると、メッセージの一時的な遅れを切断と判断してしまう可能性があるため、30

秒程度のタイムアウトの設定は妥当であると考える。

3.2.7 あとがき

モバイルコンピューティング環境におけるビデオ会議やビデオ電話などのヒューマンコミュニケーションを支援する同期・非同期通信方式を提案した。クライアント、会議サーバとホームサーバというシステム構成をとり、協同作業のような共有情報を扱う会議においては共有のサーバに蓄積する集中型、ビデオ電話のような個人的情報を扱う会議では各自のホームサーバに情報を蓄積する分散型の情報の蓄積を行う。

ユーザが発行する参加形態要求と通信路の接続状態により、ユーザの会議の参加形態を決定し、各ユーザの参加形態から会議形態を決定、切り替える手順を述べた。また、集中型、分散型の会議情報の蓄積方法について述べ、切断中のユーザへの送信は分散型の蓄積により、実現できることを説明した。ホームサーバがユーザの代わりに会議サーバに接続し、代理で受信、蓄積を行う。

ビデオ会議システムを実装し、同期・非同期通信方式を組み込み、部分非同期会議の会議実験を行った。会議の欠席者も後から蓄積された会議情報を閲覧することで会議進行を知ることができ、会議情報の蓄積の有効性が確かめられた。また、ユーザが同期参加できるかを判断する通信路ウォッチャの接続、切断状態の実用性を確認した。

同期・非同期通信方式の今後の課題は、ホームサーバを実装し、その実用可能性を検証することである。また、現在の手法は実際に切断してから、蓄積の開始までに遅延が存在し、会議情報の一部が落ちる問題を解決する必要がある。切断が起きてから、タイムアウトが起き、切断を検出するまでの遅延と、分散型の蓄積の場合、会議サーバがホームサーバに蓄積を依頼し、ホームサーバが蓄積を開始するまでの遅延が存在する。同期会議でも突然の切断に備えて蓄積を行うことでこの問題を解消できると思われる。

一方、ヒューマンコミュニケーションの課題として、部分非同期会議における会議のバックトラックの問題がある。同期参加しているユーザ間での合意が、非同期参加しているユーザによる反対意見で破棄され、会議が後戻りし、会議の効率が落ちる可能性がある。会議の結論をだすとき、採決をとるときは全員が同期参加する必要がある。これは、同期・非同期通信方式の問題というよりも、ビデオ会議システム側の問題であるが、採決をとるときに非同期参加のユーザに確認するような仕組みを検討する必要がある。

3.3 選択的マルチメディア通信方式と先読み方式

3.3.1 まえがき

携帯電話、PHSなどの広域無線通信インフラの発展と、ノートパソコン、PDAなどの携帯端末の高機能化、低価格化を背景に、これらを利用していつでもどこでも情報を発信、受信できるモバイルコンピューティングが注目を集めている。モバイルコンピューティ

ングの実現により、なにか伝えたい情報が発生した時、なにか情報が欲しくなった時、迅速にその要求を処理することが可能になる [4]。

その情報発信、受信を、ビデオ（動画）、ボイス（音声）、オーディオ（音楽）、イメージ（静止画）などのマルチメディアを利用して行いたいという要求がある。現在、インターネット上ではビデオ会議、電話、ラジオ、ビデオ放送などのマルチメディア通信アプリケーションが次々に登場している。しかし、広域無線通信を利用するモバイルコンピューティング環境でやりとりされるマルチメディア情報は、低解像度のイメージ、低品質のボイスに制限されている。

これは、無線通信環境でマルチメディア情報を扱おうとすると、まずマルチメディア情報を転送するのに必要とされるだけの転送速度が得られなかったり、ランダム誤り、バースト誤りなどの誤りの増加や移動時のハンドオフより、転送が途切れるといった障害が起きるためである [56]。現在、PHS が 29.2Kbps、携帯電話が 28.8Kbps のデータ通信サービスを提供をするなど、以前に比べ高速化がなされているが、依然、マルチメディア通信に十分な転送速度とはいえない。また、転送の途切れが起こるとビデオ再生がぎくしゃくしたり、オーディオ再生の途切れが生じる。ビデオ、オーディオなどの連続メディアの通信を行うには、転送するマルチメディア情報が必要とする転送速度と、途切れず、スムーズな転送を提供する通信路が必要である。すなわち、通信サービスの保証が要求される。しかし、無線通信ではこの要求を満たすのは難しい。

この通信サービス品質が保証されない問題に対し、インターネットでは変動する通信サービス品質に適応する手法が提案された。インターネットも輻輳や不可の増加により、スループットの低下、パケットロスの増加、遅延の増加などの障害が起きる。そこで、スループット、遅延などの通信サービス品質を監視し、マルチメディア情報の空間的、時間的解像度を調節して転送すべきマルチメディアデータの容量を削減したり、バッファリングや再生点の調節などを行って、パケットの転送遅延の変動を吸収する [19][20][17]

しかし、適応的手法はマルチメディア情報のコンテンツを考慮せずに単純に品質を落とす処理を行うので、重要な情報が欠落してしまうことがあるという欠点がある。また、ある大きさまでの遅延の変動の吸収はできるが、転送の途切れが長く続くと再生を続行できなくなる。無線通信環境はインターネットに比べより激しく品質が変動することが予想される。

本節では、低速で不安定な通信環境において、蓄積型のマルチメディアコンテンツに効率的にアクセスすることを可能にする選択的マルチメディア通信方式 SMAP（Selective Multimedia Access Protocol）と先読み方式を提案する。

マルチメディアコンテンツの作成者あるいは提供者は、各シーンの意味的重要度に基づいて優先度を与える。SMAP は、その優先度に基づいた選択的転送、先読み、キャッシングを行う。測定によって予測されるスループットを越えない範囲で、優先度に基づいた選択的なマルチメディア転送を行う。低速通信環境でも高優先度を与えられた重要な情報を落とすことなく、ユーザに提供する。また、高優先度の情報をあらかじめ先読みしてストレージに保持しておくことにより、転送の途切れが一時的に生じても再生を続行させる。さらに、キャッシングにより、巻きもどし時のマルチメディア再生の品質を向上させる。

以下、3.3.2で関連研究について述べ、3.3.3で選択的マルチメディア通信方式 SMAP と先読み方式を提案する。3.3.4では両方式の実装について述べ、3.3.5で性能評価について報告する。

3.3.2 関連研究

2.4.3、2.4.3で無線通信、インターネットのマルチメディア通信の技術を紹介し、2.4.3で本論文は、インターネットのマルチメディア通信技術に基づくエンドエンド制御のビデオ転送の通信アーキテクチャを採用することを述べた。選択的マルチメディア通信方式と先読み方式はインターネットのマルチメディア通信技術の適応的手法に属するものである。

以下で、選択的マルチメディア通信方式、先読み方式それぞれについて関連研究を挙げる。

選択的マルチメディア通信方式

利用している通信路において、スループットが低下したり、パケットロス率が上がったら、通信路の転送能力以上のデータを送りこまないように転送データ量を減らす必要がある。さもないとパケットロスがより増加し、ビデオ、オーディオパケットの遅延が大きくなって再生タイミングに間に合わなくなり、マルチメディア再生品質が低下する。そこで、転送するビデオ、ボイスデータ量を増減する「スケーラブルメディア」の技術が開発され、ビデオ会議システムやビデオ配信システムに組み込まれた [19][20][21][22]。フレームレートや画像のサイズ、色数を変えたり、階層的符号化 [23] を利用して転送する階層を増減させたり、要求ビットレートの異なる圧縮符号化方式を切り替えて利用することで、転送データ量を調節する。

しかし、マルチメディア情報のコンテンツを考慮せずに、機械的に品質調節処理するという問題がある。重要な情報は品質を低下させずに、ユーザに提供したい。

QoS 指定を含むシナリオ記述言語 [57] は、あるシーンのビデオ、オーディオの時間的解像度、空間的解像度がどれだけ低下してもよいかの許諾指定、そのシーンのビデオ、オーディオ間の優先度の指定などを記述できる。この記述に従って、ビデオかオーディオを優先したり、ビデオの動き（時間的解像度）とビデオフレームの画質（空間的解像度）のどちらかを優先してそのシーンの再生を行うことができる。例えば、ゴルフのレッスンビデオにおいてインストラクターがお手本のスイングを見せるシーンでは動きを優先し、説明のシーンではボイスをビデオよりも優先する。この手法は、シーン内の再生制御を扱うといえる。

選択的マルチメディア通信方式は、シーン1よりもシーン2の方が重要な場合、シーン1の情報量を落として転送するかわりに、シーン2の情報量を増やすというシーン間の再生制御を行うことが特長である。コンテンツのシーン中で相対的に重要な情報をより高品質にユーザに提供する。

アプリケーションレベルのビデオやオーディオなど連続メディアのリアルタイムの配信（ストリーミングと呼ばれる）プロトコルとして、RTSP(Real Time Streaming Protocol)[58]

が Internet-Draft として提案されている。RTSP に準拠することで、クライアント・サーバ型マルチメディア製品同士のインターオペラビリティ（相互利用）が期待できる。RTSP は、セッションの確立やライセンス処理など高レベルの制御機構を提供する。

選択的マルチメディア通信方式はビデオフレーム、オーディオブロックの転送スケジューリングやバッファ管理を行うもので、マルチメディアアプリケーションのミドルウェア的位置づけである。

先読み方式

先読みに関しては、ビデオオンデマンドシステムにおいて VBR(可変ビットレート) のビデオの余剰帯域を利用して、前もってユーザのバッファにビデオデータを入れる手法が提案されているが [59]、提案する先読み方式は、先読みの一部の帯域を割り当てて、常に先読みを行う。先読みの一部の帯域を割り当てることには、フレームレートなどのマルチメディア再生品質が平均的に低下し、先読みした情報を保持するストレージが必要になるという欠点があるものの、マルチメディア再生が途切れたり、停止するより好ましいと考える。

また、WWW のようなハイパーテキストやデータベースにおいてユーザが次に見るであろうデータを先読みしておくことで、ユーザの待ち時間を減らす為の先読み方式 [60] が提案されている。提案する先読み方式は、選択的マルチメディア通信方式と組み合わせることで高優先度をもつ重要な情報を先読みすることができる。

3.3.3 選択的マルチメディア通信方式と先読み方式

本節で選択的マルチメディア通信方式 SMAP と先読み方式の提案を行う。本方式の対象とするマルチメディアデータは特に符号化方式は問わないが、そのデータが要求するビットレートは対象ネットワーク環境で予想されるスループットの 2,3 倍程度が望ましい。あまりに要求ビットレートと実際のスループットの差が大きいと、たくさんのビデオフレームやオーディオブロックのスキップされ、マルチメディア再生に支障が起きる。

基本方式

まず、マルチメディアコンテンツへの優先度付けについて説明する。マルチメディアコンテンツはビデオとオーディオを含み、ビデオは時間軸に沿って連続するビデオフレーム、オーディオはオーディオサンプルから構成されるとする。本方式ではオーディオをサンプル単位でなくブロック単位で扱う。1 ブロックは 0.05 秒の長さのオーディオサンプルから構成される。

マルチメディアデータの作成者、配布者は、シーンの意味的な重要度に基づいてシーン間で相対的な優先度を 4 段階 (0-3) で与える。シーンとはある時間範囲の連続したビデオフレーム群、オーディオブロック群を指す。シーンへの優先度の与え方のポリシーは各優先度データ作成者の主観にゆだねられる。例えば、コマーシャルビデオでの場合、商品の説明のシー

コンテンツ指向の優先度

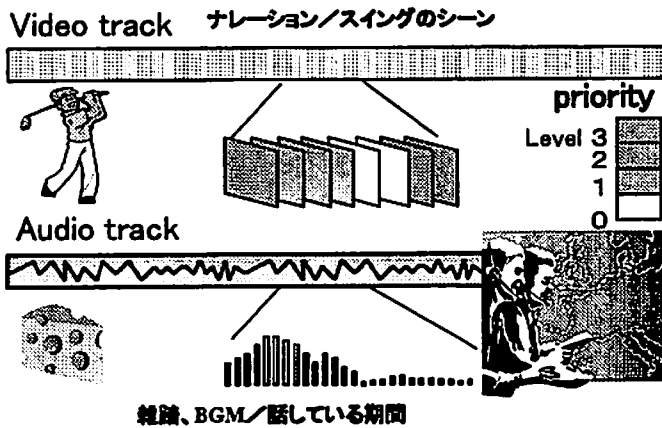


図 3.10: マルチメディアデータへの優先度付け

ン、ゴルフのレッスンビデオの場合、インストラクターがお手本のスイングを見せるシーンはその他のシーンよりも高い優先度が与えられるだろう。ニュースビデオの場合、キャスターを撮すシーンよりも事件、事故の様子のシーンに高い優先度を与えるだろう。3.3.4では優先度の作成を支援するツールについて述べる。

以下で、マルチメディアデータへの優先度付けに基づく選択的転送、先読み、キャッシングの3つの手法を説明する。

選択的転送

測定により予測されるスループットを越えない範囲で、優先度の高いビデオフレーム、オーディオブロックを選択的に転送する。上記のようにマルチメディアコンテンツのシーンに対し優先度がつけられているので、優先度の低いシーンでは、ビデオフレーム、オーディオブロックの転送量が少なくなるかわりに、高優先度のシーンのフレーム、ブロックが多く転送される。本節の後ろの方で、選択的転送のスケジューリングアルゴリズムについて述べる。

先読み

誤り、ハンドオフ、あるいは輻輳により転送が一時的に途切れても、再生を続行することができるように、情報の先読みを行う手法を提案する。帯域幅の一部を先読みのために割り当て、常に再生点からある時間分の情報を保持すれば、保持している時間分の転送の途切れを吸収できる。アルゴリズムの詳細は、以降の選択的転送スケジューリングの項で述べる。選択的転送と先読みを組み合わせることで、高優先度のフレーム、ブロックを先読みすることができ、転送が途切れても重要な情報はユーザに提供されることができる。

また、先読み方式ではロスしたパケットの再送を提案する。インターネットの連続メディアの実時間通信では普通、再送を行っても再送パケットが再生のタイミングまでに間に合わないことから、再送が行われることは少ない。しかし、先読みされたパケットは再送する時間的余裕がある可能性がある。ビデオフレームは複数のパケットに分割して配送され、そのすべてのパケットが受信者に到着しなければ元のビデオフレームを再構成することはできない。再送により、再生品質を向上することができる。また、先読みを選択的転送と組み合わせて用いた場合、先読みされるフレーム、ブロックは優先度が高く、重要だとマーク付けされたものであり、再送を行う意味がある。

高優先度情報のキャッシング

受信者端末のストレージの許す限り、高優先度のビデオフレーム、オーディオブロックをキャッシングする。低速通信環境の場合、ユーザにはダイジェストのように高優先度のみが提示される。ユーザはより詳細な情報が欲しい場合、任意のタイミングで再生の巻き戻しを行うことができる。すでに取得した高優先度のフレーム、ブロックは受信者端末に蓄積されているので、未取得の低優先度フレーム、ブロックを獲得し、併せて再生することで、より高品質のマルチメディア再生を提供することができる。

現在のサブノート PC などの携帯端末は GB 単位の容量の大きなハードディスクを装備しており、現在の無線通信環境で扱うような比較的低品質のマルチメディア情報ならば十分な容量であるといえる。キャッシングの詳細は 3.3.4 節で述べる。

選択的転送スケジューリング

選択的転送と先読みを実現するスケジューリングアルゴリズムについて述べる。マルチメディア情報の送信者は、このスケジューリングアルゴリズムに基づき、ビデオフレーム、オーディオブロックを選択し、受信者に送信する。

まず、前提として、予測スループット B (bps)、エンドツーエンドの予測遅延 D (秒) が与えられると仮定する。これらのパラメータはマルチメディア情報の受信者のパケット監視、測定により、一定時間ごとに更新される (3.3.4 参照)。また、ユーザは再生開始時、送信ビットレートのビデオとオーディオに対する分配の割合、通常転送と先読みの割合を指定するものとする。

送信者は、送信ビットレートを B (bps) に設定する。つまり、送信者は予測されるスループットを超えない範囲でパケットを送信する。前述したとおり、 B は一定時間ごとに更新される。ユーザの指定した送信ビットレートのビデオ、オーディオの分配割合に従って、ビデオの送信ビットレートを B_V 、オーディオの送信ビットレートを B_A に設定する ($B = B_V + B_A$)。さらに、ユーザの指定した通常転送と先読みの分配割合に従って、通常ビデオ転送、先読みビデオ転送、通常オーディオ転送、先読みオーディオ転送の送信ビットレートを $(1 - \alpha)B_V$ 、 αB_V 、 $(1 - \beta)B_A$ 、 βB_A に設定する。 α, β はビデオの先読みの割合、オーディオの先読みの割合であり、 $(0 \leq \alpha, \beta < 1)$ である。図 3.11 に送信ビットレートの分配を示

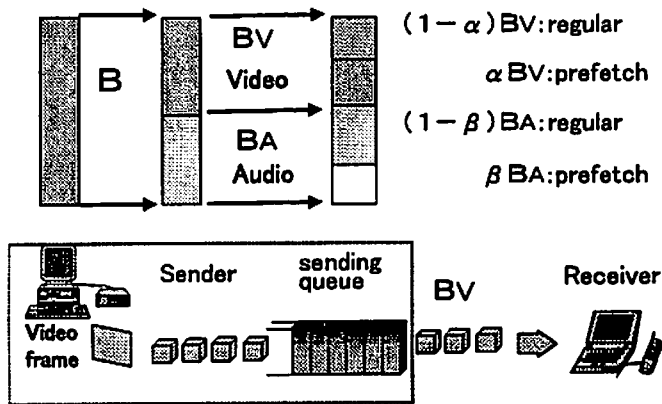


図 3.11: 送信ビットレートの分配

す。

マルチメディア情報の送信側はビデオ、オーディオそれぞれに独立して送信キューを持ち、送信ビットレート B_V, B_A のレートで、送信キューからパケットを取り出し転送する。各送信キューが空の時に、以下で説明する選択的スケジューリングが実行され、パケット化されたビデオフレーム、オーディオブロックが送信キューに入れられる。選択的スケジューリングには、通常スケジューリングと先読みスケジューリングの2種類のモードがあり、ビデオの場合 α 、オーディオの場合 β の割合で、先読みスケジューリングが行われる。

ビデオにおける選択的スケジューリングアルゴリズムを図 3.12 に示す。オーディオの場合も同様のアルゴリズムが用いられる。

通常スケジューリングにおいては、(マルチメディア情報の受信側の現在の再生点 $N +$ 予測ネットワーク遅延 D) の時刻のフレームから、スケジューリング対象時間 S 後までの範囲の未送信フレームを送信の対象とする。その中からもっとも優先度が高く、先頭に近いフレームを選択し、パケット化し、送信キューに入れる。 S (スケジューリング対象時間と呼ぶ) を大きくすると時間的に先の情報も送信の対象となることから、高優先度フレームが積極的に選ばれる。一方、 S を短くすると優先度を考慮しない通常の転送になる。

先読みスケジューリングにおいては $(N + D + S)$ から $(N + D + S + P)$ までの範囲の未送信フレームを送信の対象とする。通常スケジューリングと同様に、その中から優先度順にフレームが選択され、送信キューに入れられる。先読み時間 P の大きさは P と同じく優先度の評価に影響する。

3.3.4 実装

図 3.13 に SMAP のアーキテクチャを示す。3.3.3 節で述べた3つの手法は、マルチメディア情報の送信側と受信側でミドルウェアとして実現される。選択的転送と先読みは送信側の SMAP Sender、高優先度情報のキャッシングは受信側の SMAP Receiver で実現される。

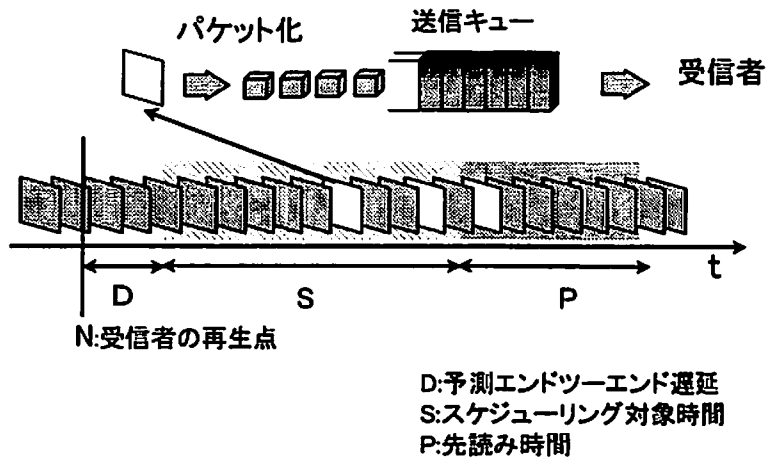


図 3.12: 選択的スケジューリング

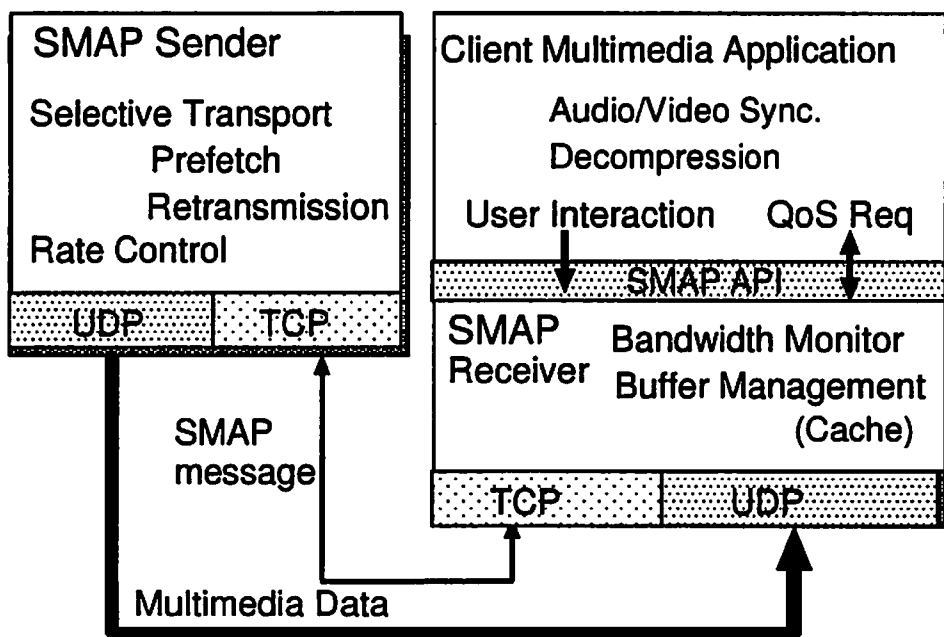


図 3.13: SMAP アーキテクチャ

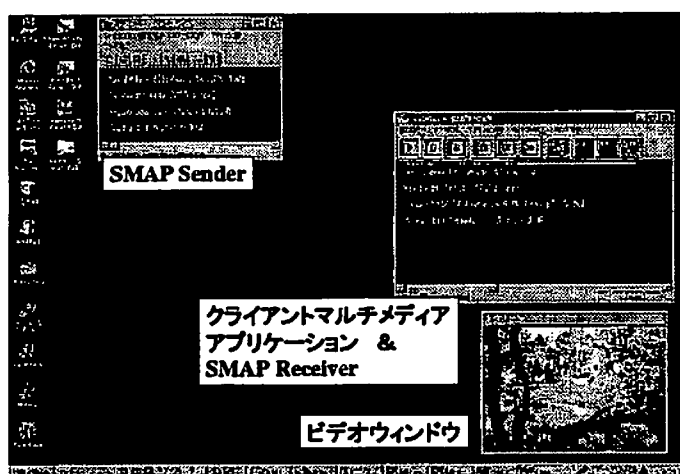


図 3.14: SMAP プロトタイプ

Windows95 上にマイクロソフトの Video for Windows 開発キットを利用して、SMAP Sender, SMAP Receiver、クライアントマルチメディアアプリケーション、そしてマルチメディアデータに優先度を与える作業を支援するツールを実装した。図 3.14に、SMAP プロトタイプの画面写真を示す。C++ 言語によりプログラムを記述し、SMAP Receiver とクライアントマルチメディアアプリケーションのステップ数はあわせて 7102、SMAP Sender のステップ数は 4285 であった。本プロトタイプは Windows 標準の AVI(Audio Video Integrated) 形式のマルチメディアデータを扱うことができ、Radius の Cinepak、Intel の Indeo などパソコン用の比較的処理コストの小さい圧縮方式が利用できる。

マルチメディアコンテンツの視聴者は、端末上でクライアントマルチメディアアプリケーションを起動する。提示されるメニューから視聴する情報を選択すると、クライアントは対応するビデオ配信サーバ (SMAP Sender) に接続する。SMAP Sender は選択的転送によってマルチメディアデータを送信し、SMAP Receiver はそれを受信し、クライアントマルチメディアアプリケーションが再生を行う。早送り、巻き戻し、一時停止などを行うことができる。

SMAP Sender

SMAP Sender はマルチメディアデータをストレージから読み出し、3.3.3節で述べたスケジューリングアルゴリズムに従った選択的転送、先読み、レート制御を行う。

SMAP Sender はある一定時間 (実装では 5 秒に設定) ごとに SMAP Receiver に要求メッセージを送り、SMAP Receiver から応答を受け取る。要求の送信時刻と応答の受信時刻から、SMAP Sender と SMAP Receiver 間のラウンドトリップ時間が算出される。このラウンドトリップ時間の半分を 3.3.3の選択的スケジューリングで用いるエンドツーエンドの予測遅延 D (秒) とする。要求メッセージは SMAP 制御要求メッセージ、応答メッセージは SMAP 制御応答メッセージと呼ばれる。

SMAP 制御応答メッセージには SMAP Receiver が測定したスループット情報、パケットの再送要求が含まれている (以下の SMAP Receiver 参照)。スループット情報により、選択的スケジューリングで用いる予測スループット B (bps) の情報が更新される。また、SMAP Sender は再送要求されたパケットが含むフレーム、ブロックの再生タイミングが予測遅延 D よりも遅ければ、再送が間に合うかと判断し、再送する。

SMAP Receiver

SMAP Receiver はパケットを受信し、ビデオフレーム、オーディオブロックを再構成した後、メモリ、ハードディスクなどのストレージに保存する。ストレージの空きがなくなったとき、もっとも低い優先度のビデオフレーム、オーディオブロックを破棄し、新しいものを收容するバッファ管理を行う。ユーザはキャッシングに割り当てるストレージの容量と、再生点から何時間前までの情報をキャッシングの対象とするかを指定する。例えば、128Kbps の映像データに対し、10MB のストレージがキャッシングに割り当てられ、15分前までの情報がキャッシングの対象時間と指定されたとする。15分間の長さのこの映像データの容量は約 14MB であり、すべてのフレームを保持することはできないが、約 70% のフレームが優先度に基づいて選択的に保持される。

また、SMAP Receiver は入力パケットのスループットの測定、パケットロスの検出を行い、SMAP 制御応答メッセージを作成する。スループットは、前回 SMAP 制御応答メッセージを送った後から現在までに受け取ったパケットの容量をその時間で割って算出される。パケットロスは、パケットのシーケンス番号のギャップにより検出される。

クライアントマルチメディアアプリケーション

クライアントマルチメディアアプリケーションは SMAP API を通じて SMAP Receiver のフレーム、音声ブロックを保持するバッファにアクセスし、その伸張を行って提示する。メディア再生の時間軸に対する同期 (メディア内同期) を行うため、映像フレームの表示が間にあわない時、フレーム表示をスキップする。メディア間同期は今後の課題である。

クライアントは最初にユーザから以下の情報の入力を受け付け、SMAP Receiver に通知する。取得するマルチメディア情報名、予想されるスループット (14.4Kbps 以下、28.8Kbps、64Kbps、128Kbps 以上の 4 段階)、送信ビットレートのビデオとオーディオへの配分、通常転送と先読み転送への配分である。これらは SMAP Sender の転送スケジューリングに必要な情報であるから、SMAP 初期化メッセージとして SMAP Sender に送られる。また、実際の再生中に、一時停止、巻き戻しなどのユーザインタラクションが起きた場合、クライアントは SMAP 再生動作メッセージを SMAP Sender に送る。

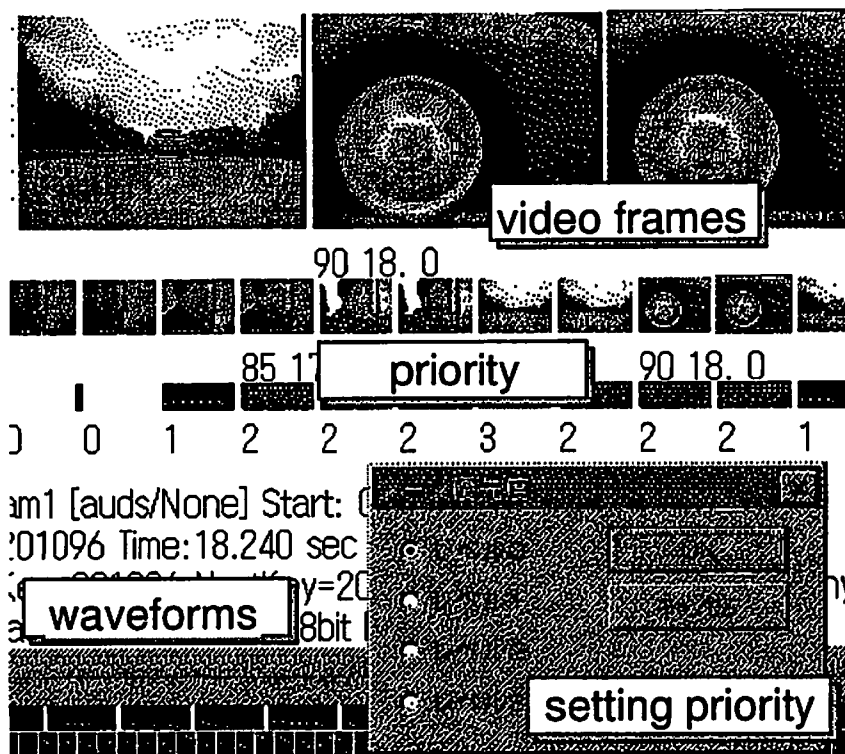


図 3.15: 優先度作成支援ツール SMAPED 画面

優先度作成支援ツール

優先度作成支援ツールは、Video for Windows の AVI ファイルを読み込み、ビデオストリーム、オーディオストリームを自動的にスキャンして、画面に縮小した数十枚のビデオフレーム、オーディオの波形を表示する。図 3.15 に開発した優先度作成支援ツールの画面を示す。マルチメディアデータの作成者、配布者などのユーザは、連続したビデオフレーム、オーディオブロック群をマウスで選択して、その選択範囲のビデオフレーム、オーディオブロックに 4 段階の優先度を与える。優先度付与が終わると優先度ファイルが生成されるので、インターネット上のサーバのストレージにマルチメディアデータと一緒に配置する。優先度ファイルはサーバの選択的転送スケジューリングで利用される。

3.3.5 性能評価

本節では、SMAP の性能評価の実験結果について報告する。選択的スケジューリング、先読み、キャッシングの有効性を評価した。あるパラメータを与えてスループットや誤りなどの通信路の状態を設定して実験を行うため、3.3.4 で開発したプロトタイプとは別に、図 3.16 のような実験システムを開発した。SMAP Sender、SMAP Receiver、クライアントマルチメディアアプリケーションが 1 つのソフトウェアとして実現されている。

実験システムはスループットの変動と、バースト誤りをエミュレートするために、SMAP Sender と SMAP Receiver の間に疑似的な通信路を設けている。実験時にスループット B(時

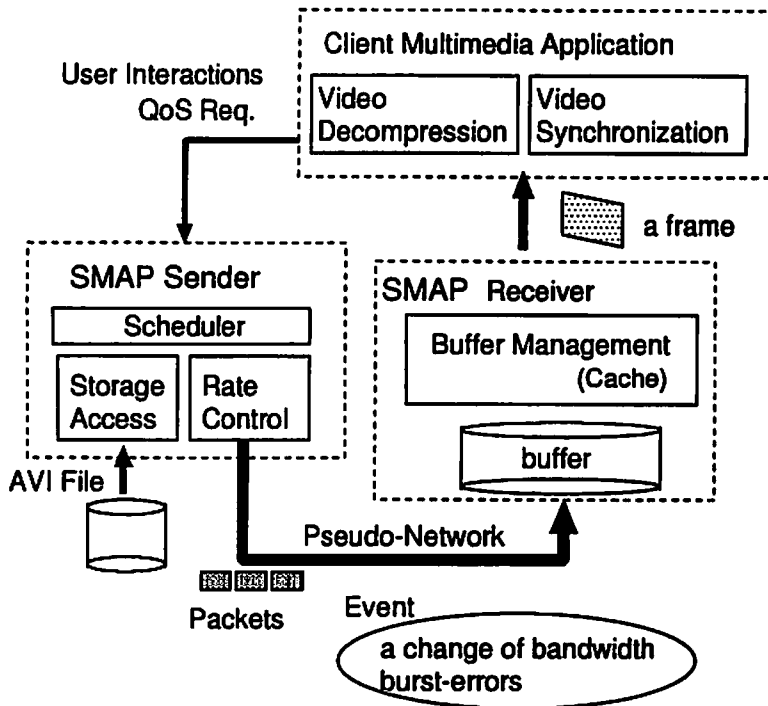


図 3.16: SMAP 実験システム

刻 t 、スループット b)、バースト誤り E (時刻 t , 長さ l) のイベントを疑似ネットワークに与えることができる。例えば $B(0s, 29.2Kbps)$ はスループットの初期設定が 29.2Kbps、 $E(30s, 15s)$ は疑似ネットワークが起動してから 30 秒後に、15 秒の長さのバースト誤りが起きたということを示す。

テストデータとして、2 分 51 秒の長さのある映画の商業ビデオ (Windows AVI 形式のファイル) を用いた。ビデオトラックとオーディオトラックが含まれるが、実験ではビデオトラックのみを扱った。ビデオ品質は、解像度 160 ドット \times 120 ドット、24 ビットカラー、フレームレート 5fps である。符号化方式は Intel Indeo で要求ビットレートは、約 150Kbps である。ビデオトラックに対し、優先度作成支援ツールを用いて優先度を与えた。優先度は、キャストの紹介のシーン、ダンスをしているシーンなどに高優先度を与え、動きの少ないタイトルのシーンや、風景を写しているシーンなどに低優先度を与えた。

選択的転送、先読み、キャッシングを評価するために 3 種類の実験を行った。なお、本プロトタイプはメディア内同期を行いながら再生を行う。つまり、10 秒のマルチメディア情報は 10 秒で再生しようと試みる。ある時刻に再生すべきフレームの受信が完了しなかった場合、そのフレームの表示はスキップされる。

選択的転送

最初の実験は、従来の優先度を考慮しない転送スケジューリングと 3.3.3 節で述べたアルゴリズムに基づく転送スケジューリングを比較するものである。プロトタイプに $B(t=0,$

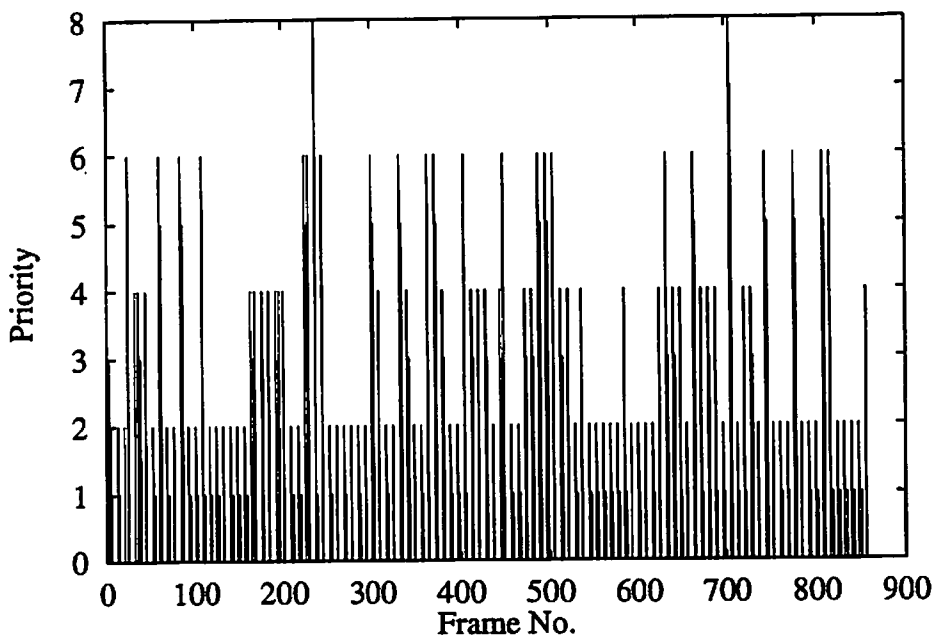


図 3.17: 従来転送

	レベル 0	レベル 1	レベル 2	レベル 3
選択的転送	11%	35%	49%	6%
通常転送	55%	27%	17%	2%

表 3.2: ユーザの参加形態の決定

$b=29.2\text{Kbps}$) をイベントとして与えた。つまり、帯域幅は 29.2Kbps 固定で、誤りは起きないという状態に設定して実験を行った。

従来方式、SMAP、それぞれの結果を図 3.17, 3.18 に示す。x 軸はフレーム番号、y 軸は優先度であり、再生されたフレームの優先度をプロットした。従来方式では周期的にフレームが選ばれ、再生されているのに対し、本方式では再生されたフレームの分布が偏っており、従来方式に比べ、より高優先度のフレームが選ばれているのがわかる。再生したフレームの優先度の分布は表 3.2 のようになっており、確かに選択的転送の方式が高優先度の高いフレームを選んでいることが分かる。

実際の再生を観察すると、従来方式は定期的に映像が更新されるのに対し、本方式は、優先度を与えたシーンはフレームレートが増加し、時間的解像度が増加する。これにより、従来手法では分からなかった人の踊っている様子や、パネルが回転する様子が確認できた。

図 3.18 はスケジューリング対象時間 S の値が 3 秒に設定された時の結果を示している。これを 6 秒に設定したところ図 3.19 のような結果が得られた。図 3.18 よりもフレームの偏りが増え、高優先度のフレームに再生が集中している。 S の大きさが確かにシーン優先度の評価に影響していることがわかる。

スケジューリング対象時間を変化させたときのフレームレートを図 3.20 に示す。5 秒間隔

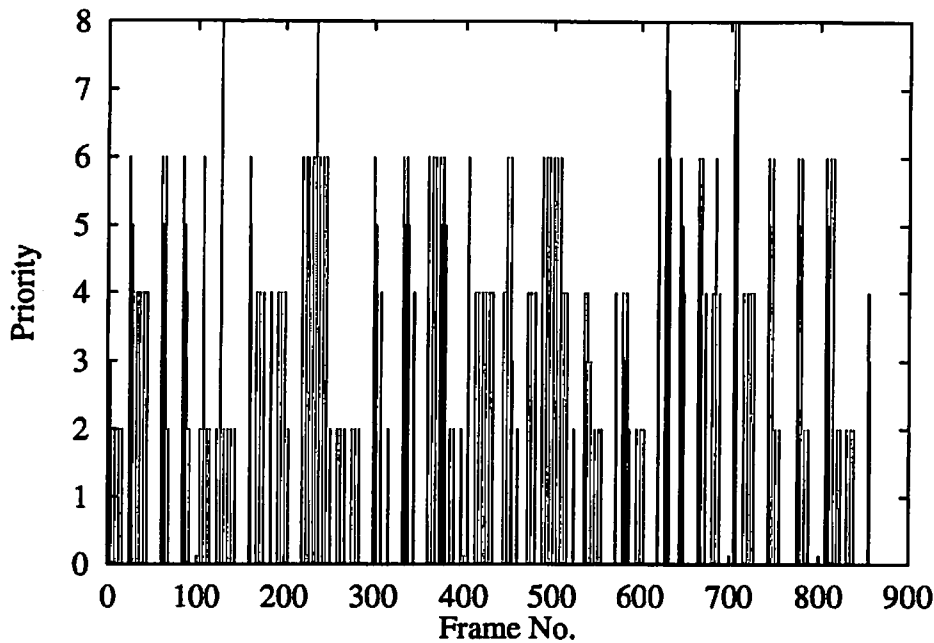


図 3.18: 選択的転送 ($F=3s$)

でフレームレートを計算し、プロットしてある。 $S = 3$ のときはフレームレートは1から2程度でほぼ一定しているが、 $S = 6$ 、 $S = 12$ では頻繁に変化している。コンテンツの種類やユーザの好みにもよるが、フレームレートのあまりに大きな変化は好ましくないため、スケジューリング対象時間 S の設定は3秒から4秒程度が妥当であると思われる。

先読み

次に、先読みを評価するためにプロトタイプに $B(0, 64Kbps)$ 、 $E(20s, 10s)$ 、 $E(50s, 15s)$ 、 $E(140s, 15s)$ というイベントを与えた。帯域幅は64Kbps固定で、時刻20秒に長さ10秒のバースト誤りが起きる。その後、時刻50秒、140秒に15秒のバースト誤りが起きる。なお、先読みには帯域幅の約40% (25Kbps) を割り当て、スケジューリング対象時間 F は3秒、先読み時間 A は10秒に設定した。

実験結果を図3.21、3.22に示す。従来方式はバースト誤りのイベントに対応して、3箇所所で映像が途切れる。しかし、先読みを行うことによって、その転送の途切れの期間にも、少ないながらもフレームの再生ができています。先読みにより、途切れる時間を短かくできることが確認できた。

キャッシング

最後に、キャッシングの評価を行うために図3.18に示した実験の後、データの先頭に巻き戻しを行って、2回目の再生を行った(図3.23)。明らかに図3.18に比べ、再生フレームが

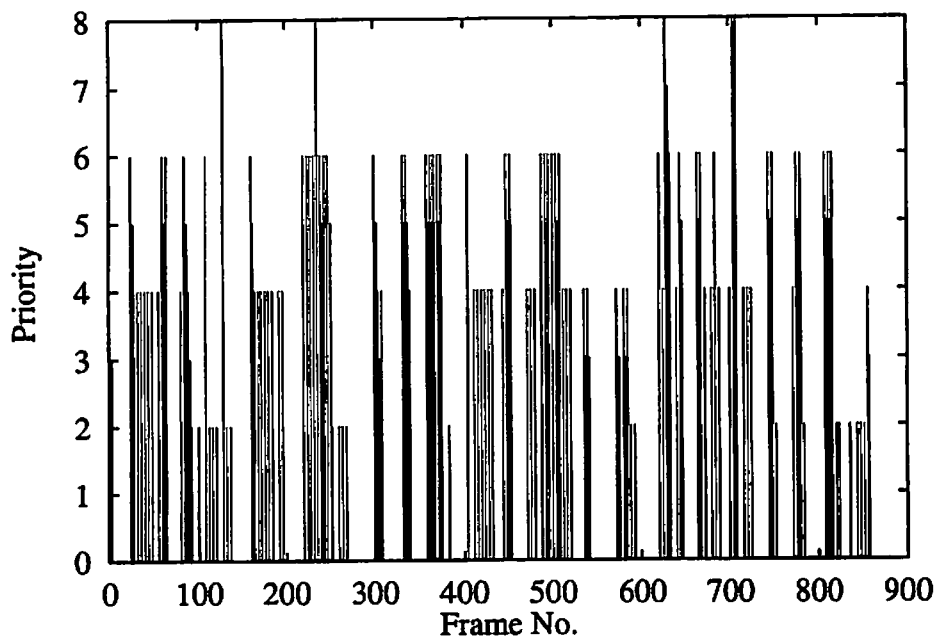


図 3.19: 選択的転送 (F=6s)

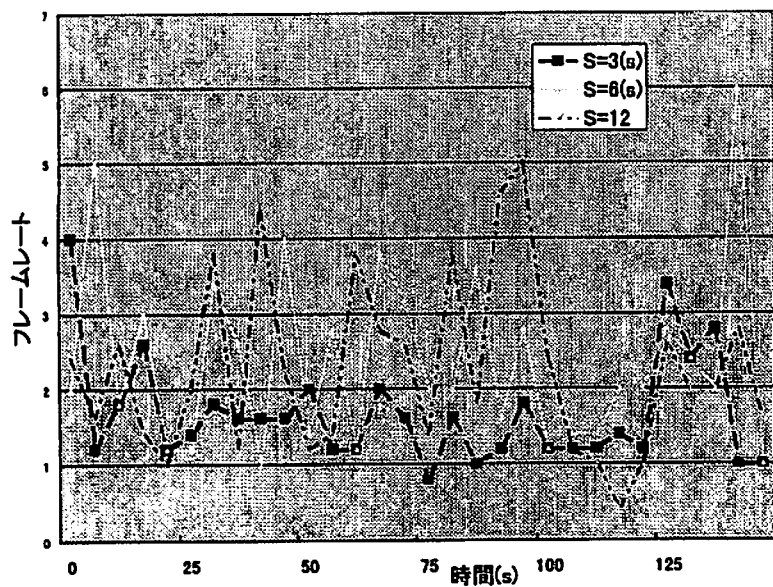


図 3.20: スケジューリング対象時間とフレームレート

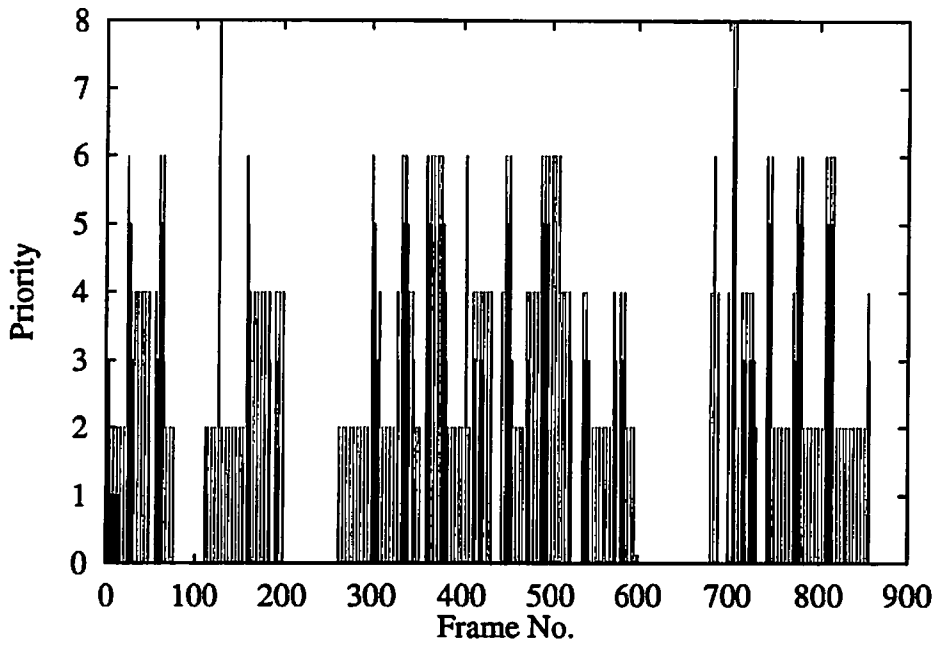


図 3.21: 先読みなし

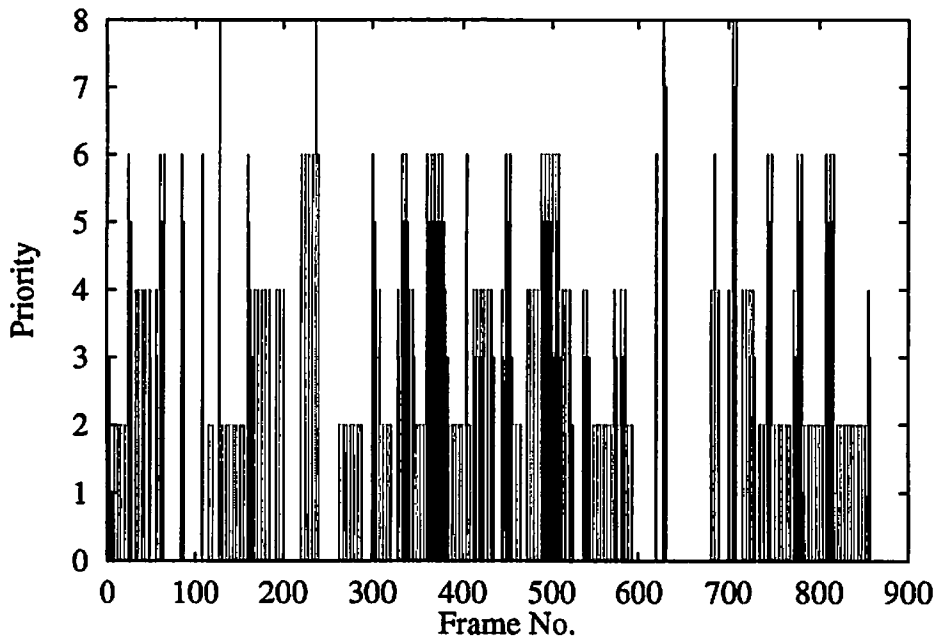


図 3.22: 先読みあり

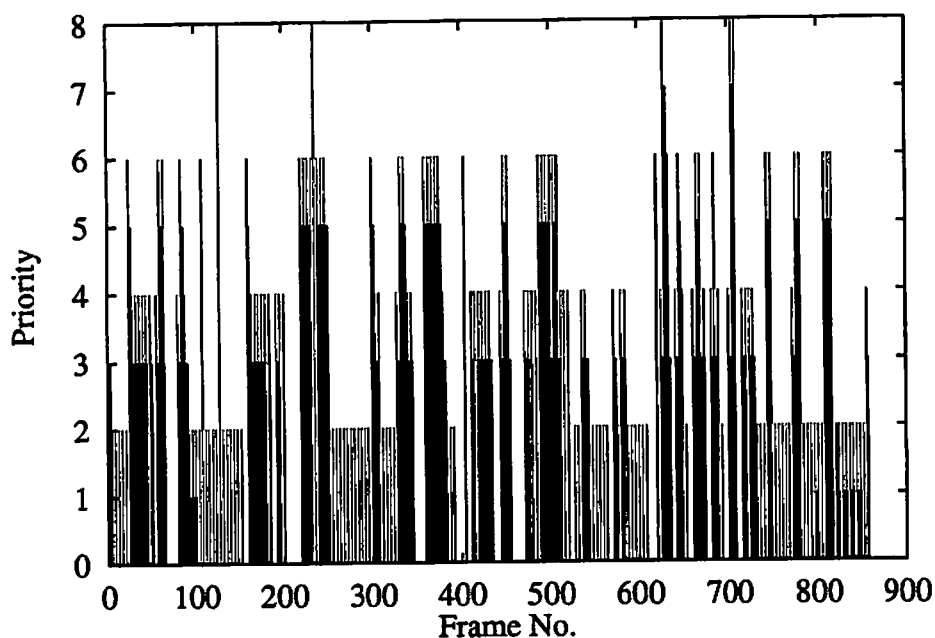


図 3.23: キャッシング

増加している。1回目の再生を補完する情報を受信しているので、再生がより本来の再生に近づくのが確認できた。2MB(メモリ 300KB、ディスク 1700KB)のバッファを用意したので、全フレーム(858枚)の約40%を収容できる計算になる。リプレイを行った本実験中に、SMAP Receiverのバッファ管理により捨てられた受信フレームは102枚であった。

3.3.6 あとがき

本節では、マルチメディアデータのコンテンツを考慮した優先度付けに基づく、選択的転送、先読み、キャッシング手法を提案した。また、選択的転送スケジューリングアルゴリズムや本方式のアーキテクチャ、実装について述べた。さらに、プロトタイプによる実験により本手法の有効性を確かめた。選択的マルチメディア通信方式は、低速回線においても重要な情報を落とすことなくユーザに提供することができ、先読みにより転送が途切れたとしても再生を続行できることがわかった。蓄積型のビデオコンテンツの取得に対し、両方式は有効であるといえる。

選択的マルチメディア通信方式は優先度付与の作業をユーザに負担させるという欠点があるが、それに対して優先度作成支援ツールを開発した。今後、このツールにビデオのシーン自動検出機能を組み込み、より優先度付与の作業を簡単にする。

選択的スケジューリングアルゴリズムのスケジューリング対象時間の設定に関してもさらなる解析、検討が必要である。実験ではスケジューリング対象時間を短くするほど、フレームレートの変動が小さくなることは確かめられた。しかし、最適な値はまだ分かっていない。

また、現在の選択的マルチメディア通信方式は、優先度に基づいてビデオフレーム、オーディオブロックを選択して送信するもので、時間的解像度の制御を行っている。すなわち、重要でないシーンのビデオフレーム、オーディオブロックを落とす代わりに、重要なシーンのフレームレートを上げ、オーディオブロックの再生量を増やしている。これに対し、低優先度のビデオフレームの空間的解像度（ビデオフレームサイズ、色数）を落として、転送する方式について検討したい。

3.4 実時間指向優先転送方式

3.4.1 まえがき

現在、インターネット上でラジオ放送やビデオ会議などを行うマルチメディア通信アプリケーションが普及しつつあり、マルチメディア通信が一般的なものになっている。音声や映像などのマルチメディア情報は時系列的に変化する連続的データであり、そのデータ転送の時間制約を保証しないと音声、映像のとぎれといった障害が起きてしまう。また、ユーザインタラクションメッセージ（マルチメディア情報再生の一時停止、早送りメッセージなど）は快適なユーザインタフェースを提供するためには、ある指定時間以内に配送されることが望ましい。アプリケーションやシステムを制御するメッセージについては、低い遅延で転送される必要がある。そこで、「要求する所定時間以内でのデータ転送を保証するサービス」、すなわちタイムクリティカル通信 (Time Critical Communication:TCC) [61] サービスが要求されるが、現在のパケットネットワークは時間制約を必要としない通常のデータ転送を対象として設計されているため、その要求に応えることができない。

これに対する現在の一般的なアプローチは、時間制約を保証しない遅延の変動あるネットワークに対しアプリケーションが適応することである。例えば、映像通信においては、フレーム落ちや、同じフレームを表示することによって遅延に順応できる。音声通信では遅延の変動に対して、音声の再生点を遅らせたり、バッファを用意して遅延の変動を吸収することによって会話が実現できる。しかし、適応にも限界がある。あまりにも遅延変動が激しい場合は会話を続けることは難しくなる。

もう一つのアプローチは、ネットワークを時間制約を保証するように改良する手法であり、ネットワークバンド幅、バッファ領域などの資源予約に関して数多くの研究が成されている。インターネットストリームプロトコル ST-II[62] はアプリケーションが申告するデータフローの特性に応じて資源をパス上で予約することによって、ストリームデータに対して効率的なストリーム転送サービスを、安定した遅延と転送レートで提供する。XTP[63] は ST-II と同様に資源を予約して転送サービスの品質を保証すると共に、強力なマルチキャスト機能、レート及び制御フロー制御機能を持つ。また、RSVP[14] は資源を予約するための専用プロトコルであり、柔軟な資源予約方法を提供している。

ただし、以下のような問題があると考えられる。サービスを保証するためには必然的に資源をあらかじめ予約しなければならないが、予約処理はパス上の送受信ホストだけでなく中間のルータ群でも行われ、それらすべてが許諾するまでネゴシエーションを続ける必要がある。

予約が完了するまで通信エンティティは通信を開始できない。また、予約した資源が実際にはあまり使われなかったり、必要以上の資源が予約されてしまったり、あるセッションが不当に資源を占有するなどして、ネットワークの利用効率が落ちる可能性がある。また、資源予約は「送受信ホスト及び中間のルータ群にはすべて提案プロトコルが実装されている」という大きな仮定をおいている。閉じたネットワークならば問題はないが、パス上に提案プロトコルが実装されていないマシンが一つでもあった場合、うまく動作しないという欠点がある。

本論文では実時間指向優先転送方式を提案する。これは、既存の IP パケットネットワークに対し改良をほどこし、実時間制約を考慮する転送サービスをインターネット環境で提供するものである。ベストエフォート（最善努力型）の概念に基づき、データ転送の時間制約の転送を完全に保証することはできないが、従来よりもより品質の保証の程度を向上させる。

実時間指向優先転送方式は、現在インターネットで広く用いられている IP [37] をベースにしたネットワーク層のプロトコルである、ネットワーク応答特性プロトコル (NPP : Network Protocol with Performance) [64][65][66] を導入する。NPP は、パケットスケジューリングや他のオーバーヘッドを減らす工夫をすることによってタイムウィンドウ内のパケット転送をできる限り保証する。パケットスケジューリングアルゴリズムは、パケットの時間制約と優先順位付けにより送受信するパケットを選択するものである。

NPP はインターネットのエンドエンドのノードとその中間のノードに配備される。インターネット環境の場合、経路上のすべてのホスト、ルータに提案プロトコルが実装されていることを仮定するのは難しい。NPP は、すべてのノードに配備されなくても動作することを考慮して設計されている。

3.4.2 内外の研究状況

タイムクリティカル通信

マルチメディア情報通信や、ファクトリオートメーション (FA) の制御用通信の分野の要求を背景として、タイムクリティカル通信 [61] の標準化が始まっている。タイムクリティカル通信には以下の 3 つの時間制約 (タイムウィンドウと呼ばれる) のタイプがある。

- TW_W : データ転送に関する時間制約 (one way time window)
- TW_R : データ転送とその応答受信までに関する時間制約 (round trip time window)
- TW_T : タイムクリティカル通信トランザクションに関する時間制約

TW_R と TW_T には、受信 TCCE が応答メッセージを作成する時間が含まれているので、アプリケーションを実時間で制御するようなメカニズム (例えば実時間 OS を利用するなど) が必要とされる。実時間指向優先転送方式は TW_W のみを扱う。図 1 は 3 つのタイムウィンドウのタイプと、タイムウィンドウ TW_W がマルチメディア通信アプリケーションで実

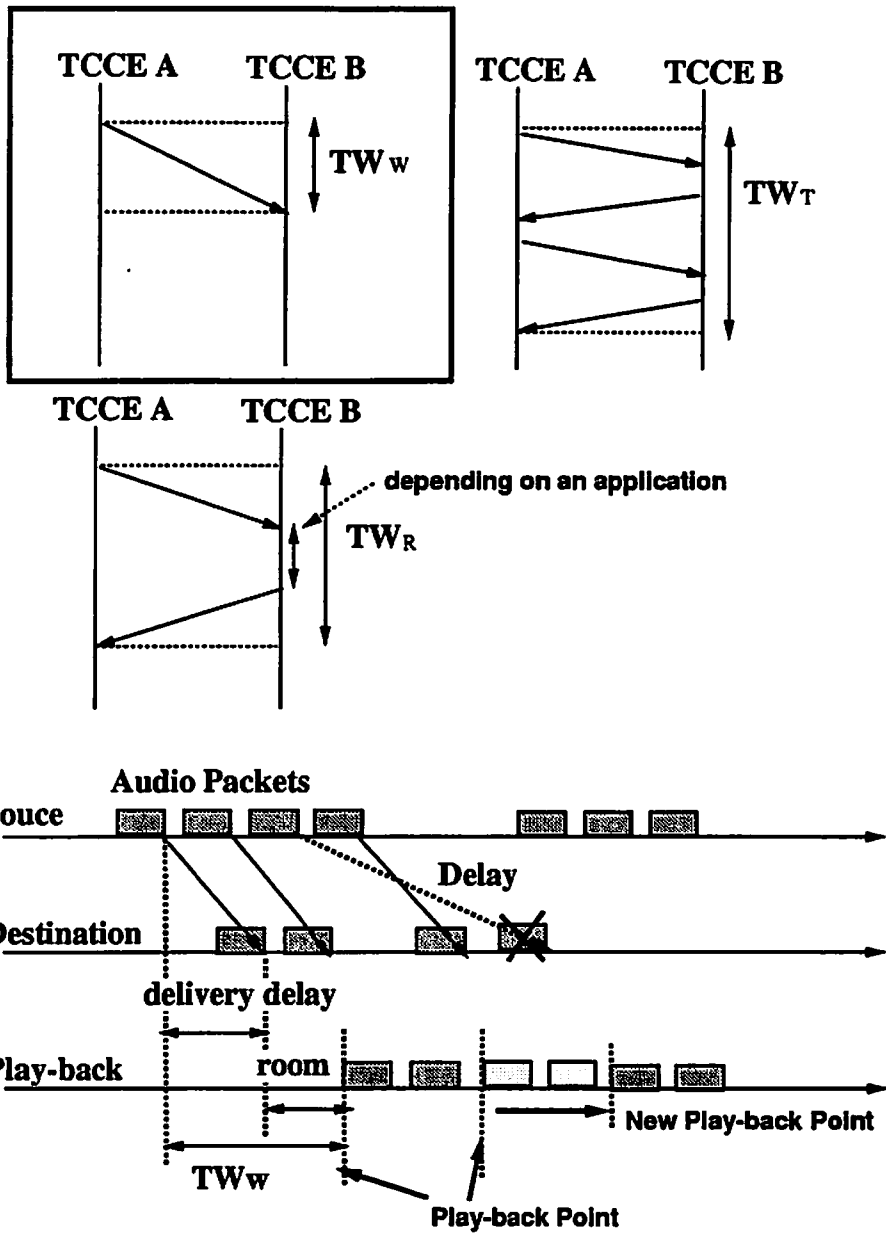


図 3.24: タイムウィンドウ

際にどのように指定されるかを示すものである。ある程度のジッタは、(適応的)アプリケーションが再生点を調整することにより吸収されることができる。

現在、インターネット環境ではビデオ会議を行える *CU - SeeMe*、ラジオ放送など音声リアルタイムで転送できる *RealAudio*、インターネット上で電話ができる *NetPhone*、オーディオだけでなくビデオも送れる *StreamWorks*(ただし、ほとんど静止画のように見える)などといったマルチメディアアプリケーションが開発され、利用されている。しかし、現在のインターネットは低速で、遅延の変動も激しい。この状況に適応するために、いずれもバッファに余裕をもたせて遅延の変動を吸収したり、再生点を遅らせるなどの工夫をしているが、オーディオの途切れやビデオの乱れが頻繁に起き、スムーズな再生ができているとはいえない。パケットにタイムウィンドウ(時間制約)をもたせてその遅延の変動が抑えられれば、よりスムーズにマルチメディア情報の再生ができるだろう。

インターネットストリームプロトコル

インターネットストリームプロトコル ST-II は、インターネット上のエンドツーエンドの保証された通信サービスを提供する、ネットワーク層のプロトコルである。データ転送レートの保証や遅延の安定性などを要求する実時間アプリケーションに対し、効率的なパケットストリーム転送サービスを提供する。

アプリケーションはストリームを作成する際に、それが要求するストリームの特性を FlowSpec として指定する。FlowSpec は、バンド幅、遅延、信頼性などのパラメータから成り、目標値と最低限の許容値の両方が指定される。すると、ST-II プロトコルはそのストリームの経路上において、リソース(バンド幅、バッファ空間)を予約する。これにより、データパケットを低遅延で転送し、ネットワークの混雑によるパケットロスの起こる確率を低くすることができる。FlowSpec フォーマットは図 3.25 のように定義されている。例えば、DutyFactor は、要求したバンド幅が実際に利用される比率を時間的に見積ったものである。ErrorRate は誤り率、Precedence はコネクションの優先順位、LimitOnPDURate は始点が許諾できる最低パケットレート、MinBytesXRate は始点が許諾できる最小バンド幅、AccdMeanDelay はデータパケットのトータルの予測ディレイ(ms) AccdDelayVariance はパケットの転送遅延の分散、DesPDUBytes は望ましい PDU サイズ、DesPDURate は要求する PDU レートというように定義されている。

図 3.26 は、A から B と C にストリームを流すようなマルチキャストの通信例である。このとき、資源予約は以下のように行なわれる。ホスト A の ST Agent が CONNECT メッセージを ST Agent 1 に送り、ST Agent 1 は A から Agent 1 までのフローのための資源を予約する。ST Agent 1 は CONNECT メッセージを B と、C に送る。ST Agent B は CONNECT メッセージを受けとり、Agent 1 と Agent B の間のフローにおいて資源を予約する。ST Agent C も B と同様に資源を予約する。資源予約時に要求するだけのネットワーク資源が利用可能でなかった、つまり余っていなかった場合は、優先度の低いストリームの接続を切ることによりその要求を受け入れようとする。優先度の低いストリームがなかったり、どうやっても受け入れることができない場合は、REFUSE メッセージが前のホッ

0	8	16	24
PCode	PBytes	Version=3	0
DutyFactor	ErrorRate	Precedence	Reliability
Tradeoffs		RecoveryTimeout	
LimitOnCost		LimitOnDelay	
LimitOnPDUBytes		LimitOnPDURate	
MinBytesXRate			
AccdMeanDelay			
AccdDelayVariance			
DesPDUBytes		DesPDURate	

図 3.25: FlowSpec フォーマット

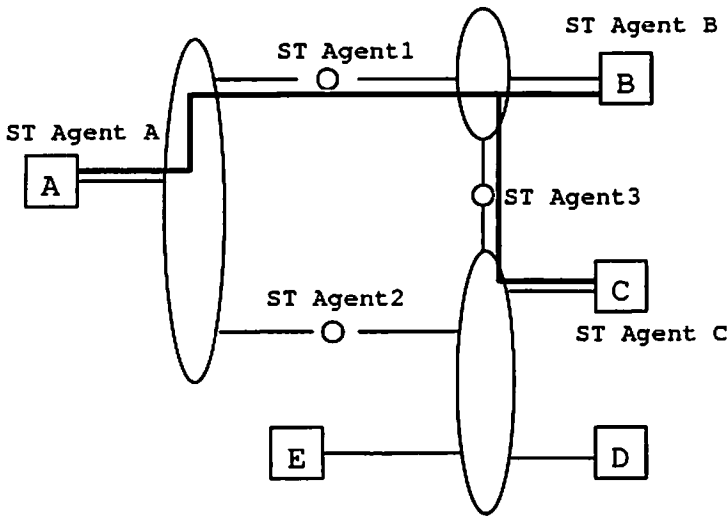


図 3.26: ST-II の資源予約

プに返される。

一部の重要なアプリケーション(医療関係、重要な会議、工場ラインなど)は資源を予約して高品位な情報転送を確実にを行う必要がある。しかし、資源予約には、以下のような問題がある。

1. 通信エンティティはすぐに通信を始めることができない。

サービスを保証するためには必然的に資源をあらかじめ予約しなければならないが、予約処理はパス上の送受信ホストだけでなく中間のルータ群で行われ、さらにそのすべてが承諾をするまでネゴシエーションを続け、予約が成功した後ついに通信エンティティが通信を始めることができる。そのセッションが長く継続する場合はあまり問題にならないが、短時間のセッションである場合はそのオーバーヘッドは大きい。

2. ネットワーク資源が効率的に利用されない可能性がある。

予約した資源が実際にはあまり使われなかったり、必要以上の資源が予約されてしまっ

たり、あるセッションが不当に資源を占有するなどして、ネットワークの利用効率が落ちる可能性がある。

3. 資源予約は「送受信ホスト及び中間のルータ群にはすべて提案プロトコルが実装されている」という大きな仮定をおいている。

クローズドなネットワークならば問題はないが、パス上に提案プロトコルが実装されていないマシンが一つでもあった場合、うまく動作しないという欠点がある。インターネットでの利用を前提とした場合、この仮定をすることは難しい。

上でも述べたように資源予約は医療関係などクリティカルな性質を持つアプリケーションでは必須である。しかし、一般的なユーザ、一般的なマルチメディアアプリケーションは必ずしも厳密な再生を要求しないと考える。本質的に資源予約を必要とするアプリケーションのみが資源予約を行うべきである。

現在のルータの QoS 制御技術

近年、多くのルータメーカが QoS 機能の強化をはかっている [67]。従来のルータには優先制御機能程度しかもっていません、可変長パケットを扱うことから固定長セルを扱う ATM スイッチのように遅延の揺らぎや使用帯域幅などをきめ細かく保証することはできなかった。ボイスをインターネット上で流したい (Voice over IP) 等のニーズに対し、プロバイダが帯域保証などの QoS サービスを実現するとその分、基幹回線の利用効率が下がるという問題がある。

シスコ、アビチ、ケーブルtron社などは図 3.27 のような手法で、基幹回線の利用効率を下げずにできるだけ精度の高い QoS 機能を実現しようとしている (シスコはこの機能を RSVP カットスルーと読んでいます)。ユーザがらプロバイダのアクセス回線では RSVP を利用し、プロバイダ内部や基幹部分では IP ヘッダの TOS (Type Of Service) フィールドを利用する。ユーザは RSVP を使って最小帯域幅やジッタなどのサービス品質をプロバイダに要求し、プロバイダのエッジルータは各ユーザ要求を 6 段階の優先度にマッピングして基幹回線にパケットを流す。優先度が 6 段階しかないので品質保証は大ざっぱになるが、回線利用効率を大きく低下させることはない。

シスコ社は他にもエッジ・ルータでトラフィックのシェーピングを行ったり、ルータの各インタフェースごとに WFQ (Weighted Fair Queueing) [16] を行う分散 WFQ を導入したり、IP パケットを 30 バイトの固定長の短いパケットに分割して細かい帯域制御を行うなどの工夫をして、QoS 制御に取り組んでいる。

他にも、米パケットティアの帯域制御装置 PacketShaper² は、特定のプロトコルやアドレス単位に最低限保証する帯域を設定でき、例えば H.323 の手順を識別して自動的に音声用の帯域を確保すると行ったことを可能にする。

²<http://www.net.macnica.co.jp/product/packeteer/packshap/i-shap00.htm>

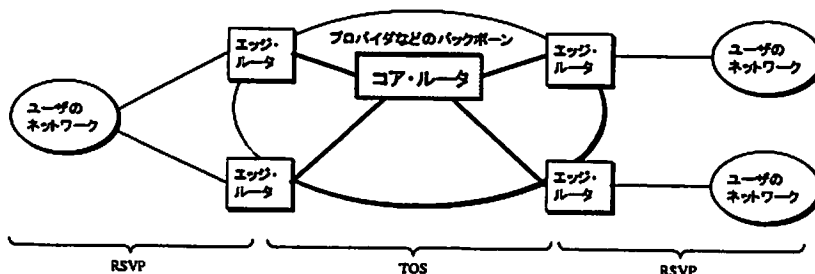


図 3.27: RSVP と優先度制御を組み合わせた QoS 制御

3.4.3 ネットワーク応答特性プロトコル

本節では、ベストエフォートのタイムクリティカル通信サービスを提供する、実時間指向優先転送方式を提案する。実時間指向優先転送方式が利用する通信プロトコル、ネットワーク応答特性プロトコル (NPP: Network Protocol with Performance) の提案も行う。また、NPP の制御プロトコル、ネットワーク応答特性モニタリングプロトコル (NPMP: Network Performance Monitoring Protocol) についても述べる。まず、基本的アイデアについて論じる。

基本的なアイデア

- オーバヘッドをおさえる：ST-II などのような資源予約を行わないことで通信前のオーバヘッドをなくす。ただし、パケットスケジューリングの処理は FIFO (First In First Out) の処理に比べ、オーバヘッドになりうる。そこで、簡易のバーチャルサーキットの構築を行う。
- IP (IPv6) との互換性：IP を置き換える形をとることによりインタネット環境で広く利用できるようにする。NPP は IP のオプションフィールドを拡張する形を取ったため、NPP パケットは NPP を実装していないルータやホストを通ることもできるし、パス上のすべてのホスト、ルータに NPP が実装されている必要もない。パス上に NPP が実装されていれば、それだけパフォーマンスの向上が期待できる。また、NPP は IPv6 [48] が標準となっても容易に対応可能である。IPv6 はフローラベルと優先度フィー

ルドを持つ。

- パケットの区別：IPにもTYPE OF SERVICEフィールドがあり、優先順位とトランスポートタイプが指定できるが、一部のホスト、ルータしか対応していないのが現状である。パケットに明示的に時間制約と優先順位を与え、これを忠実に制御するプロトコルを提案する。パケットの重要度と実時間制約は独立である。重要なパケットでも実時間制約が緩い場合もあるし、重要でないパケットでも実時間制約が厳しい場合もある。
- 時間制約を持ったパケットの消失率を下げる：リアルタイム通信においては、パケット消失時の再送が意味をなさない場合があり、メッセージ消失率を減らす必要がある。メッセージが失われる原因として以下の3つが考えられる。

1. ルータのバッファに空きがない
2. ノイズの影響でパケットが電氣的にゆがめられた
3. ネットワークの障害による経路の断絶

このうち、1については輻輳時に、優先度の低いパケットを率先して捨てることにより対応できる。2については現在のネットワーク、特に光ファイバチャネルは非常に誤りが低いので無視できるだろう。3については、データ転送を開始する間にパイロットパケットをとばすことにすれば(後述するが、NPPはデータ転送前にネットワーク応答特性(予測転送遅延)を得るためにパイロットパケットをとばす)、あらかじめ障害を発見できる。

これらのアイデアは「パケットスケジューリング」と「ネットワーク応答特性獲得機構」とで実現される。

パケットスケジューリング パケットスケジューリングは時間制約が厳しいパケットを優先して処理するものであり、時間制約の緩い(ない)パケットに邪魔されることなく配送されることができる。ホスト、ルータ内の送受信処理でパケットスケジューリングが行われる。また、パケットの優先順位も優先制御の対象とする。

ネットワーク応答特性獲得機構 ネットワーク応答特性獲得機構は、ネットワークがサービス可能なタイムウィンドウを見積もるのに用いられる。達成不可能なタイムウィンドウ要求は意味がない。始点から終点までのホップ数が増えれば一般的に転送遅延は増加し、始点と終点が同一のLANに接続していたとしてもその転送時間には限界がある。そこで、ネットワーク応答特性獲得機構は送信ホストから受信ホストまでの転送時間を測定し、送信TCCEにその情報を提供するのである。ただし、ネットワークの遅延はダイナミックに変動するので、あくまでその遅延情報は予測値である。実際にその遅延で転送されるかは分からない。このときトータルホップ数、パス上の最小MTUの情報を得て、送信ホスト上のNPPに報告すると共に、そのパス上でバーチャルサーキットを構築する。

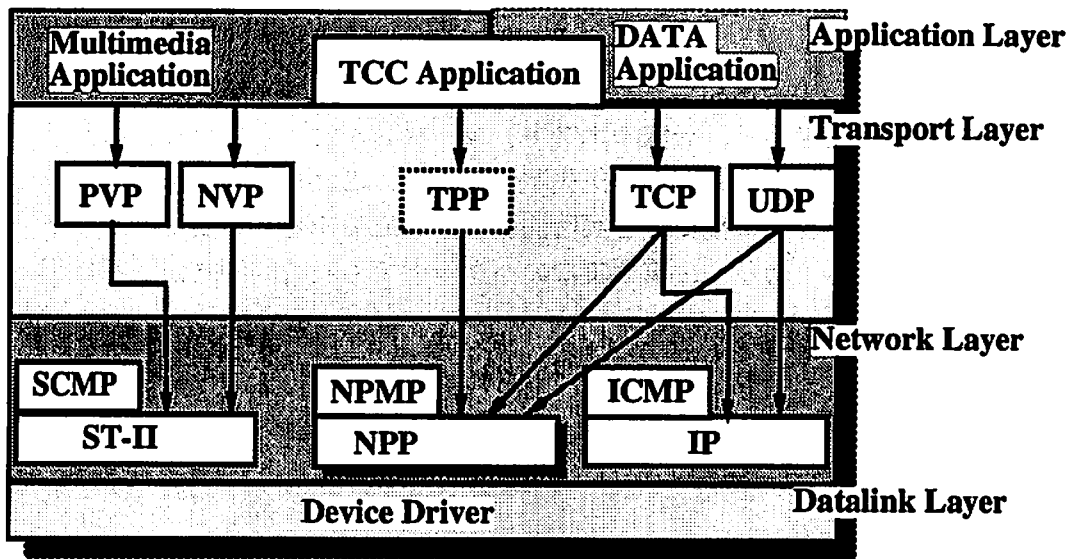


図 3.28: NPP プロトコル階層

ネットワーク応答特性獲得機構の利用は、データ送信の前の付加的な処理であり、オーバーヘッドを招く。そこで、以下の場合にはネットワーク応答特性獲得機構を利用しない。

- 以前にネットワーク応答特性獲得機構を利用したので、すでにタイムウィンドウの見積りができている場合 (ただし、そのタイムウィンドウ情報はすでに古いものになっているかもしれない)。
- パケットには優先順位のみを指定し、タイムウィンドウを指定しない場合

NPP は資源予約や入力トラフィックを制限する許諾制御の機構を含まない。これらは上位層が扱う仕事である。

プロトコル階層

本節では、NPP とそのプロトコル階層の位置付けについて述べる。NPP は IP と同様、ネットワーク層のプロトコルで、各種ネットワークを相互接続するコネクションレスのプロトコルであるが、タイムクリティカルなデータグラムを扱う点が異なる。パケットのタイムウィンドウと重要度によってパケットスケジューリングを行う。また、ネットワーク応答特性獲得機構は、ネットワーク応答特性モニタリングプロトコル (NPMP: Network Performance Monitoring Protocol) により実現される。図 3.28 は NPP のプロトコル階層を示すものである。

NPP の上位層のプロトコル TPP (TransPort Protocol for NPP) は TCC アプリケーションに対するインタフェースと、タイムクリティカルメッセージが失われた場合の処理 (再送など) を行うものである。コネクション型の通信を要求するタイムクリティカル通信アプリケーションもあるので、UDP と TCP のようにコネクション型とコネクションレス型それぞれに対応した TPP が必要である。

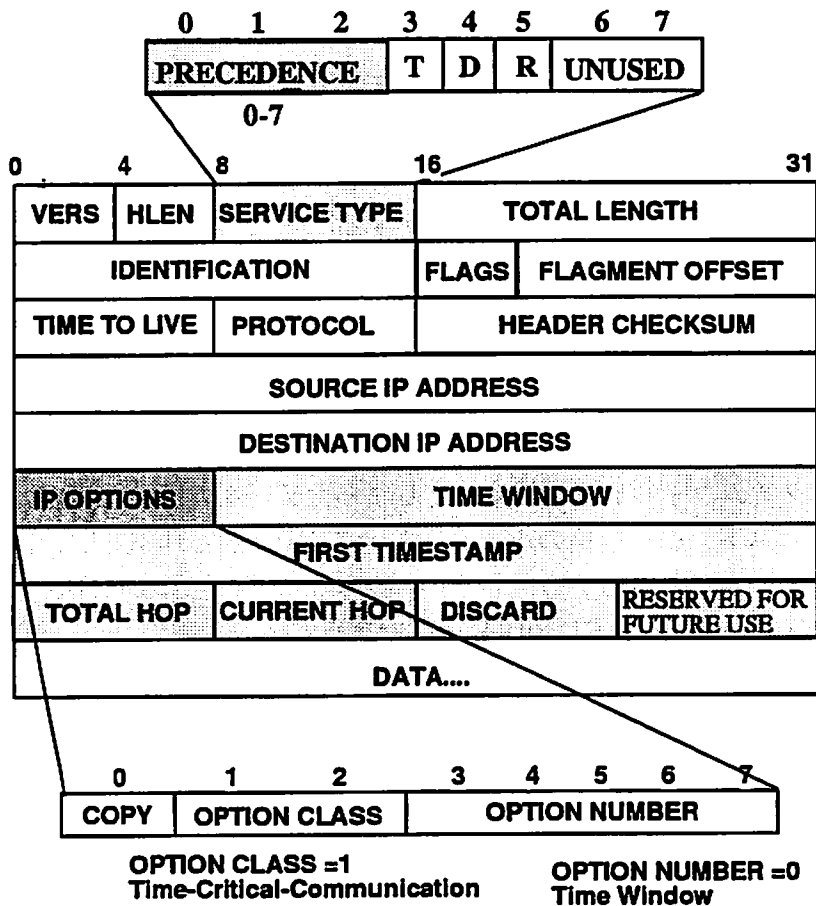


図 3.29: NPP ヘッダフォーマット

なお、NPP は IP の機能をすべて持っているので、IP と完全に置き換えが可能である。TCP、UDP をなんら変更することなく従来の TCP/IP アプリケーションが利用できる。

NPP 規定

NPP のデータグラムフォーマットは IP と互換性を保つよう図 3.29 のように設計した。各フィールドについて以下説明する。まず、PRECEDENCE フィールドは次のように定義した。パケットの種類に応じて TCCE がレベル 0 ~ 7 を指定する。

- レベル 0: 通常データ
- レベル 1: マルチメディアデータ L
- レベル 2: マルチメディアデータ H
- レベル 3: 対話型データ
- レベル 4: アプリケーションの制御データ

- レベル5以上: システムの制御データ

IP OPTIONS フィールドのオプションクラス1をタイムクリティカル通信、オプション番号0をタイムウィンドウ指定と定義した。各優先順位レベルを持つデータは、IP OPTIONS フィールドに OPTION CLASS=1、OPTION NUMBER=0を指定することにより、TIME WINDOW フィールドにタイムウィンドウ要求をms単位で指定することができる。FIRST TIMESTAMPにはTCCEが下位層にパケットを渡した時刻が埋めこまれる。TOTAL HOP フィールドは、パケットが終点までに通過するルータの数を示す(タイムウィンドウを指定するパケットは終点までのホップ数を知っている必要があるが、これはNPMPにより得ることができる)。CURRENT HOP フィールドは現在までに通過したルータの数を示す(始点で0に初期化され、ルータのパケット送出時にインクリメントされる)。DISCARD フィールドにはホスト、ルータにおいてタイムウィンドウを超過してしまったデータの扱いを指定する。つまり、送信ホスト、受信ホスト、途中経路のルータにおいてNPPが処理する際、タイムウィンドウをすでに超過していることが分かったらそのパケットを棄てるのか(DISCARD=1)、それともそのまま送るのか(DISCARD=0)を指定できる。

上位プロトコルあるいはTCCEからは、PRECEDENCE、TIME WINDOW、FIRST TIMESTAMPが与えられる。NPPはこれらのパラメータを受け取って、各フィールドを埋める。

NPPは以下のようなアルゴリズムを利用して、ルータ、ホストのバッファ内でプロトコル処理を待っているパケットの内、どれを処理するかを選択する(パケットスケジューリング)。ただし、計算機の時計はNTP[68]などによりある精度で時計合わせがされているということを仮定している。

「優先度が最も高いパケットを選択する。同じ優先度のパケットが複数あったら、時間制約(タイムウィンドウ)をもつパケットを優先する。時間制約を持つパケットが複数ある場合は、時間制約の厳しいパケットを選択する。」

CURRENT HOPとTOTAL HOPフィールド、FIRST TIMESTAMPフィールドを使用して、時間制約の最も厳しいパケットを選択する。

まず、現在の時間NOWからFIRST TIMESTAMPの値を引いたものとタイムウィンドウの比を計算する。

$$T = \frac{NOW - FIRST\ TIME\ STAMP}{TIME\ WINDOW}$$

Tは現在までに消費した時間の割り合いである。次に、CURRENT HOP+1とTOTAL HOP+1の比を計算する(1を足すのは以下の計算上のためである。CURRENT HOPをそのパケットの進んだ距離の指標としているが、パケットが最初のルータに到着した時、CURRENT HOPが0、つまり距離が0であるというのは適切ではない)。

$$D = \frac{CURRENT\ HOP + 1}{TOTAL\ HOP + 1}$$

Dは現在までに到達した距離の割り合いを近似していると考え、D/Tを時間制約の厳しさとする。D/Tが大きいということは、時間制約がゆるいことを示し、D/Tが小さいということはパケットが遅れている、すなわち時間制約が厳しいことを示す。

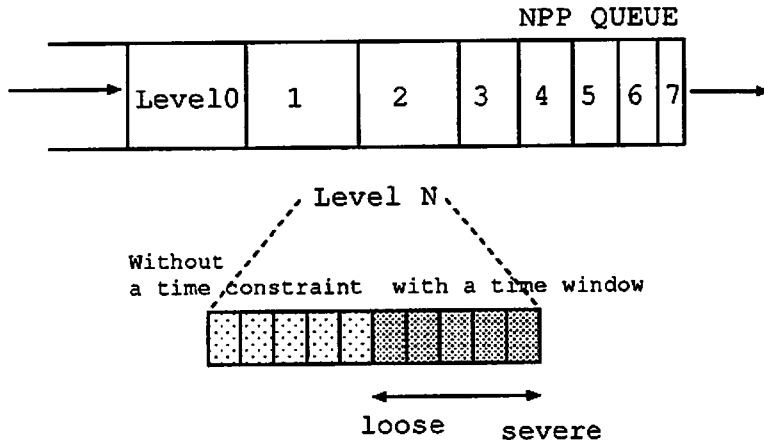


図 3.30: パケットスケジューリング

通過したルートの数 packets の進んだ距離の指標としているので、場合によってはタイムウィンドウに余裕がある方が先に選ばれてしまうことがあり、厳密には最適なスケジューリングができない。しかし、ほとんどの場合このアルゴリズムにより時間的制約の厳しいものを優先的に選択することができる。

さらにアルゴリズムに以下の機能が追加される。

- スタベーション回避

本パケットスケジューリングアルゴリズムでは優先度の低いデータはその送信を無限に延期されてしまう恐れがある。そこで、一定時間バッファに滞在した場合、そのパケットの優先度を1上げる。

- 重要パケットのロス回避

ルータにおいてバッファに空きがない場合、バッファ内のもっとも優先度の低いパケットを選んで廃棄する。制御データなど重要度が高いパケットが廃棄される確率が下がる。

- 不要パケットの廃棄

Discard オプションが設定されているパケットは、ホスト、ルータにおいてタイムウィンドウを超過していることが分かったら、廃棄される。再生点よりも遅れて到着した場合に意味をなさないビデオ、オーディオなどのマルチメディアデータに対して有効である。無駄なパケットを配送しないので、資源を有効利用できる。

NPMP 規定

NPMP は送信 TCCE がネットワークがサービス可能なタイムウィンドウを見積もることができるように、送信ホストから受信ホストまでの転送時間を測定してその情報を提供する。このときさらにトータルホップ数や、パス上の最小 MTU の情報を得て、送信ホスト上の NPP に報告する。また、そのパス上でバーチャルサーキットを構築する。送信前にパイ

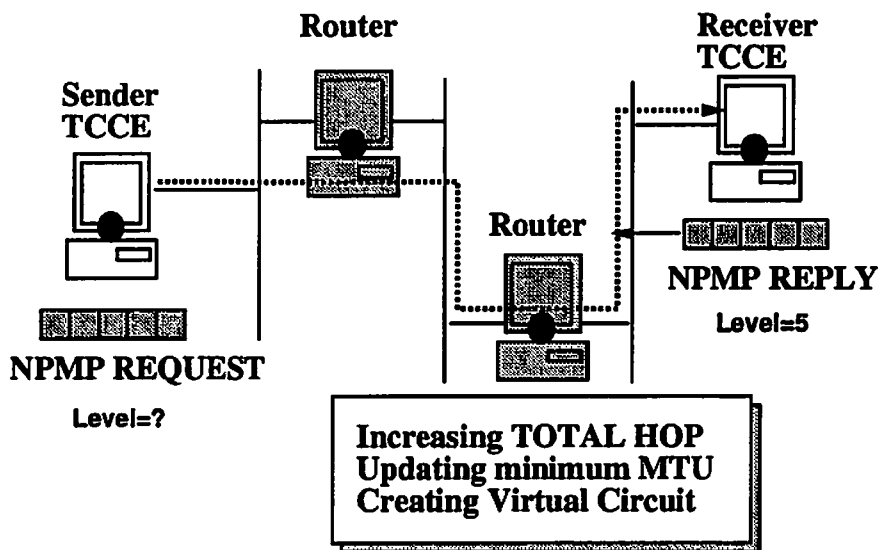


図 3.31: NPMP Mechanism

ロットパケットを飛ばすのでオーバーヘッドが増えるが、データ転送の前にそのパス上に障害がないことを確かめられるという利点がある。図 3.31は NPMP に関する手順を示すものである。

1. 送信 TCCE は NPMP に対して、受信ホストの NPMP に対し *NPMP REQUEST* を発行するよう要求する。NPP を利用している場合エンドツーエンドの遅延は優先順位の与えかたによって変わる。送信 TCCE は *NPMP REQUEST* のパラメータとして測定したい優先順位を指定する。例えば、これからユーザインタラクションメッセージを送る場合は、優先順位 3 を指定する (すると *NPMP REQUEST* の PRECEDENCE フィールドに 3 が設定される)。
2. *NPMP REQUEST* が中間のルータを通る時、TOTAL HOP フィールドが 1 インクリメントされると共に、最小 MTU フィールドが更新される (現在の最小 MTU フィールドよりも次のリンクの MTU が小さい場合)。また、付加的なルーティングテーブルに足跡を残していつて (cut-through switching と呼ばれる)、片方向のバーチャルサーキットを構築する。テーブルの管理などの詳細については本論文では扱わない。
3. *NPMP REQUEST* が受信ホストの NPMP に到着したら、その NPMP は現在の時間から TIMESTAMP フィールドの時間を引くことによってエンドツーエンドの転送遅延を計算し、その値を *NPMP REPLY* の DELAY フィールドに埋め込む。 *NPMP REQUEST* の MINIMUM MTU フィールドと TOTAL HOP フィールドもそれぞれ *NPMP REPLY* の適切なフィールドにコピーされる。
4. *NPMP REPLY* は、優先度レベル 5 で転送されるので低い遅延で送信ホストに到着することが期待できる。

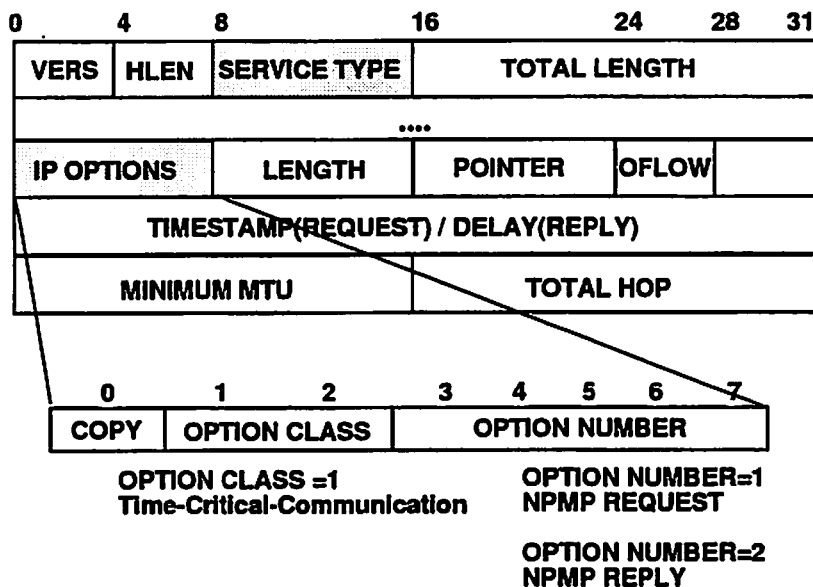


図 3.32: NPMP Header(REQUEST/REPLY)

- 送信ホストの NPMP が *NPMP REPLY* を受信し、DELAY を送信 TCCE に通達する。送信 TCCE はその値によってタイムウィンドウを決定することになる。MINIMUM MTU と TOTAL HOP については、送信ホストの NPP に報告する。始点と終点と同じネットワークに属している場合は、始点終点間の遅延とホップ数 0 という情報が TCCE に報告されるだろう。

図 3.32 に *NPMP REQUEST* と *NPMP REPLY* のフォーマットを示す。

受信ホスト、ルータに NPMP が実装されていない場合の動作について述べる。受信ホストに NPP が実装されていない場合は、*NPMP REQUEST* は受信ホストに廃棄される。送信ホストの NPMP は適切な時間だけ待って応答が返ってこない場合はタイムアウトを起こし、送信 TCCE に対し“受信ホストには NPMP が実装されていないので遅延の見積りはできません”と報告する。タイムウィンドウを見積ることはできないので送信 TCCE はタイムウィンドウ指定をすることはできず、優先順位の指定のみが可能である。一方、ルータに NPMP がない場合はトータルホップ数や最小 MTU などの情報は正しく得られない。また、バーチャルサーキットの構築もされない。このとき、以下のような 3 つの影響があると考えられる。

- パケットスケジューリングを正確に行うことができない。
- フラグメントが起こる可能性がある。
- パケットの経路が変わる可能性があるので遅延が大きく変わるかもしれない。

それでも、NPP の実装されているホスト、ルータでは優先順位によるパケットスケジューリングが行われるので、従来に比べサービス向上が期待できる。

データ転送手順

送信 TCCE はその要求や、パス上の状態によってパケット送信においてタイムウィンドウ + 優先順位指定と、優先順位指定のみの 2 通りの方法を選ぶことができる。

タイムウィンドウ + 優先順位 送信 TCCE がタイムウィンドウ要求をする場合、パス上の送信ホストと受信ホスト、中間のルータすべてに NPP 及び NPMP が実装されている必要がある。送信の手順としては、まず NPMP によりパスの遅延を獲得して要求するタイムウィンドウを決定した後、そのタイムウィンドウを指定してデータを送信するという形になる。ただし、以前の測定で遅延が既知である場合は NPMP を利用する必要はない。マルチメディア情報などのストリーミックデータを転送する場合は、遅延が時間とともに刻々と変化し、古い測定遅延の情報が意味をなさなくなってしまうので、ある周期で NPMP を用いて最新の遅延情報を得る必要がある。

優先順位のみ もともとタイムウィンドウ指定が必要ない場合や、NPMP を利用することによるオーバーヘッドを被りたくない、すなわちすぐにデータ転送を行いたいという場合は優先順位のみを指定すればよい。このとき、パス上のすべてのホスト、ルータに NPP、NPMP が実装されていなければいけないという制約はなくなる。ただし、フラグメントの回避やあらかじめパス上に障害がないことの確認、バーチャルサーキットの構築、といった NPMP による恩恵を受けることができない。

3.4.4 マルチメディア通信システム適用例

NPP の適用例として、マルチメディア通信システムへの適用を考察する。クライアントはマルチメディア情報の提示の時刻や順番などについて記したシナリオに従って次々にマルチメディア情報の転送要求をサーバに送る。サーバはその要求に対応したマルチメディア情報を流すとする。このシステムで流れるデータの種類は以下の 4 つである。それぞれの優先順位は 3.4.3 で定義した優先順位レベルに従って割り当てた。

- ビデオデータ: レベル 1 (マルチメディアデータ L)
- オーディオデータ: レベル 2 (マルチメディアデータ H)
- ユーザインタラクション: レベル 3 (対話型データ)
- 制御メッセージ: レベル 4 (AP の制御データ)

人間は一般的に、ビデオよりもオーディオの障害に敏感であり、少しの再生の途切れに対しても許容できない。そこで、オーディオの優先順位を高くしている。ユーザインタラクションとは、マルチメディア再生の一時停止、早送りや、シナリオ内の分岐選択などのデータを指す。制御メッセージとは、サーバに対する送信要求、レート制御などのメッセージである。

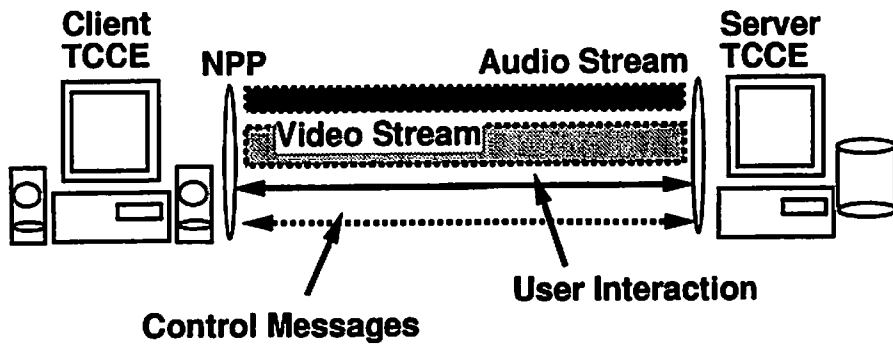


図 3.33: マルチメディアシステム構成

このシステムがNPPを利用する状況を考える。TCCEとNPPの間のトランスポートプロトコルについては検討中であり、本論文ではTCCEが直接NPPにアクセスとする。また、このマルチメディアシステムは優先順位レベル1～4それぞれについて、サーバからクライアントまでの遅延時間 $D_1 \sim D_4$ といった情報をNPMPによりすでに獲得しているとする。

ビデオデータが192Kbpsのレートを必要とするマルチメディアデータであった場合、TCCEは優先順位レベル1の遅延時間 D_1 をタイムウィンドウとして、優先順位レベル1を与えたパケットを192bpsのレートでNPPに渡す。もちろん、パス上の空き帯域が小さく元々192Kbpsのデータを送れる余裕がないとか、オーディオデータや他トラフィックの通信による影響で192Kbpsのレートが守れない場合が予想される。このため、クライアントはビデオデータの受信レートを測定し、これが現在の送信レートを下まわるようであれば、サーバに送信レートを落すように制御メッセージ(優先順位レベル4)を送るようにする。制御メッセージは高い優先度で送られるので、ネットワーク状態に対する迅速な対応が期待できる(制御メッセージが送られてからタイムウィンドウ D_4 以内に、サーバはレートを下げる)。また、ビデオデータがその再生開始点よりも遅れて到着することは意味がないので、サーバの送信アプリケーションはDiscardオプションを指定することもできる。

3.4.5 シミュレーションによる性能評価

本節では、図3.34のようなシミュレーションモデルにより、NPPの性能をIPと比較する。試験トラフィック(Test Traffic)は送信ホストSourceで生成され、ルータRouterを通過し、宛先ホスト最後にDestinationで受信される。ルータは試験トラフィックと共に負荷トラフィック(Load Traffic)を受ける。試験トラフィック、負荷トラフィックは制御データ、マルチメディアデータ、通常データの3つのデータから成る。シミュレーションは以下のような条件で実施された。

- 制御データ、マルチメディアデータ、通常データの到着過程はポアソン過程に従う。
- すべての生成されるパケットは固定長で、ホスト、ルータにおいて一定のサービス時

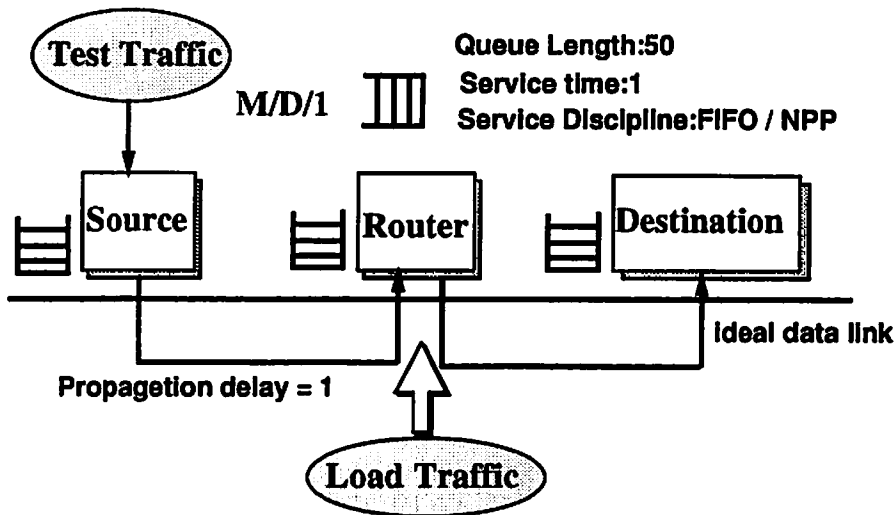


図 3.34: Simulation Model

間で処理されるとする。NPP のサービス時間のほうがパケットスケジューリングの分、IP よりも時間がかかると思われる。しかし、ここでは単純に両方とも 1(クロック)とした。到着率は負荷と等しくなる。試験トラフィックのパケットは、Source, Router, Destination においてプロトコル処理されるので、そのパケットのサービス時間の合計は 3 (クロック)となる。

- IP の待ち行列のサービス規律は FIFO、NPP は 3.4.3 で定義したパケットスケジューリングアルゴリズムを利用する。
- NPP、IP の位置するネットワーク層の下位層は理想的データリンク層であると仮定する。つまり、ビットの誤りや、イーサネットのような衝突が起こらず、バンド幅も無限であるとする。よって、トラフィックのスループットや遅延などはネットワーク層によってのみ制限される。2つのマシンのデータリンク間の伝播遅延をここでは 1(クロック)と仮定した。試験トラフィックのパケットは伝播時に合計 2(クロック)を被ることになる (Source-Router 間と Router-Destination 間)。
- すべてのパケットの DISCARD オプションを 0 に設定した。NPP のキュー内でパケットがタイムウィンドウを超過しても、廃棄されずに必ず終点に届けられる。また、スタベーションの回避制御は行わない。
- 制御データとマルチメディアデータのタイムウィンドウは簡単のため、それぞれ単純に 7(クロック)、8(クロック)に設定した。本来は、NPMP を利用することにより、アプリケーション自身が要求するタイムウィンドウを決定する。試験トラフィックが被るサービス時間と伝播遅延の合計は上記のとおり 3(クロック)、2(クロック)であるので、Source から Destination までの最小遅延は NPP、IP のキュー内での遅延がまったくなく、キューに入ったらすぐにサービスされて送信される場合であり、5(クロック)

data type	Test Traffic	Load Traffic
	arrival rate(Load)	
Control data	0.1	0.1
Multimedia data	0.3	0.0 - 0.4
Normal data	0.1	0.2

表 3.3: シミュレーションパラメータ

ク)である。

- シミュレーションではルータの数を1つとした。

試験トラフィックと負荷トラフィックは、表 3.3にしたがって生成される。負荷トラフィックのマルチメディアデータの負荷をパラメータとして、シミュレーションを行った。負荷トラフィックのマルチメディアデータの負荷を0から0.4まで変化させて転送遅延を計測した。

図 3.35は、IP と NPP の平均転送時間を示すものである。IP はすべての種類のパケットを公平に扱うので、マルチメディアデータの負荷の増加に伴って、どのパケットの転送時間も増加する。制御データ (CT)、マルチメディアデータ (MM)、通常データ (NM)、どれも同じ転送遅延を被る。

しかし NPP の場合はマルチメディアデータの負荷にもかかわらず、制御データの平均転送時間は、ほぼ一定に抑えられる。これはパケットスケジューリングの効果によるものである。

一方、図 3.36は、転送サービスの保証の確率を示したグラフである。タイムウィンドウ内に転送されることができたパケットの割合を示す。NPP は85%以上の制御データ (CT)の通信サービス保証を達成している。しかし、マルチメディアデータ (MM)の時間制約を守ることはできていない。むしろ、IP に比べると転送時間は増加している。NPP は制御データを優先するために、マルチメディアデータの転送を遅らせているためである。NPP は入力トラフィックをある一定の量に制限するような許諾制御機構も持っていないので、負荷とともに遅延が増加する。

このシミュレーションは、パス上の中間ルータの負荷の変化がパケットの転送時間に対しどのような影響を与えるかを調べたものである。表 3.4は負荷トラフィックの増加に対する、平均転送時間の増加量を示している。ネットワークの負荷が低い範囲 (マルチメディアデータの負荷が0.05-0.10に変化) と高い範囲 (負荷が0.20-0.25) について調べた。遅延の増加量の元の転送時間に対する割合を括弧の中にパーセンテージで示している。例えば、マルチメディアデータの列の左側の行の0.53(8.42%) というのは負荷トラフィックのマルチメディアデータの負荷が0.05-0.10まで増加した際、平均転送時間は0.53(クロック)増加し、その増加分というのは負荷が0.05だった時の平均転送時間の8.42%に相当するという意味である。高負荷時には、遅延の増加が激しくなることが分かる。

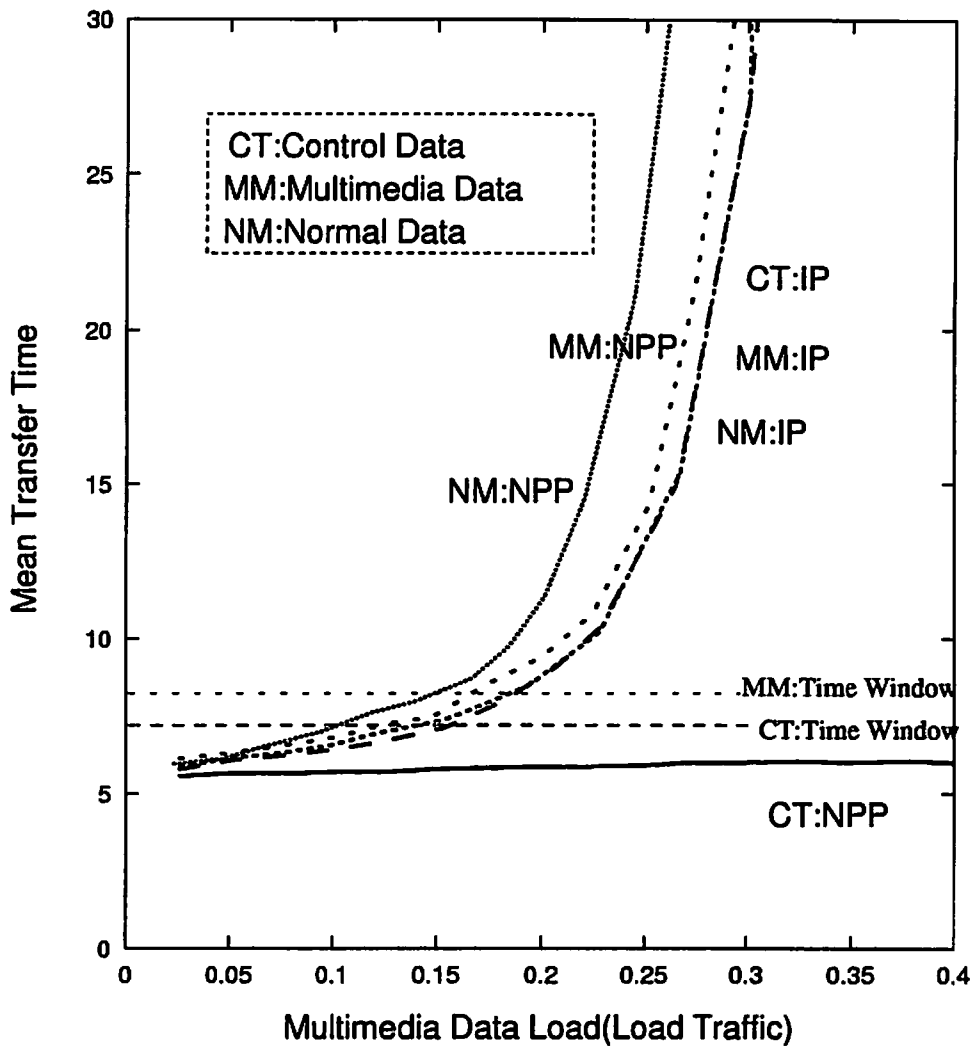


図 3.35: 負荷と平均転送時間

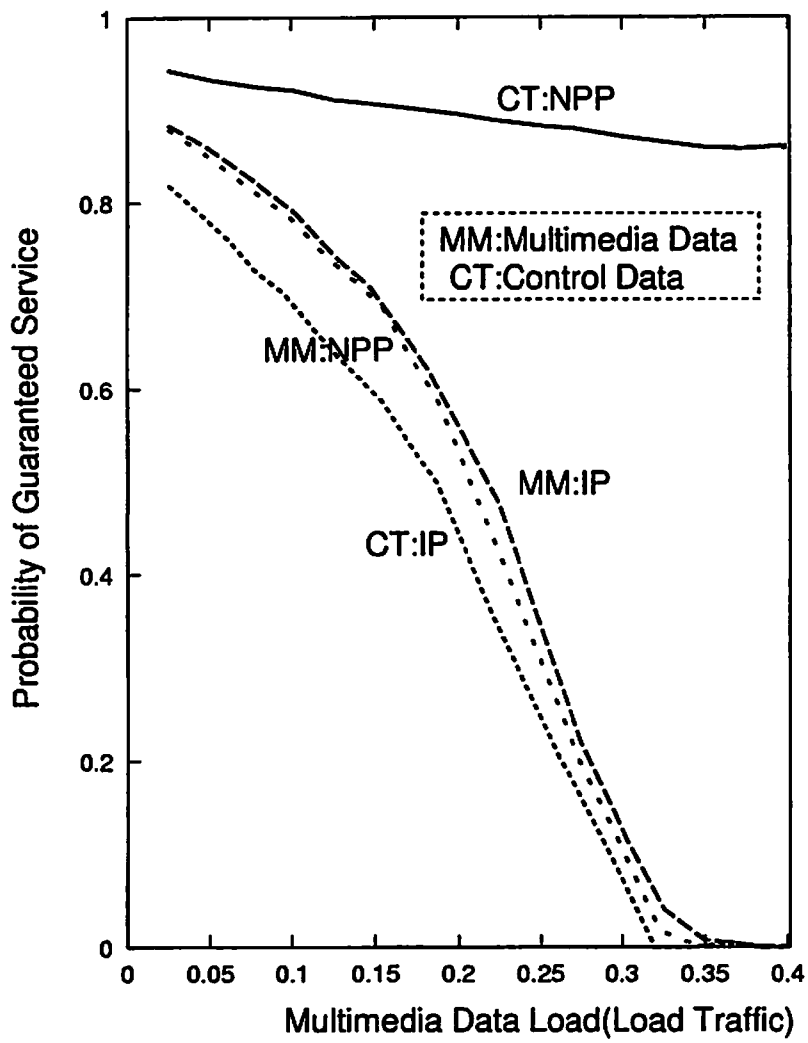


図 3.36: 負荷とサービス保証確率

Test Traffic Data Type	Multimedia Data Load	
	0.05-0.10	0.20-0.25
Control data	0.05(0.89%)	5.86 - 5.91(0.85%)
Multimedia data	0.53(8.42%)	4.9(52.18%)
Normal data	0.57(8.61%)	12.26(110.15%)

表 3.4: 平均転送遅延の増加量とその割合

以上のシミュレーションによって、NPP の特性に関して以下のような知見が得られる。 T_n を優先順位 n のトラフィックとする。

あるパスにおいて T_n が増加すると、 $T_l (l < n)$ の遅延はかなり増える。低い優先順位のトラフィックは高い優先順位のトラフィックの犠牲となる。 T_n の遅延も増加するが、その割合は T_n 自身の量に依存する。 T_n の量が少ない場合はそれほど遅延は増加しないが、 T_n の量が多い場合、負荷の増加量が同じでも、遅延の増加の割合が大きくなる。一方、優先度の高いトラフィック $T_h (h > n)$ はほとんど影響を受けない。遅延はほぼ一定である。

3.4.6 BSD/OS への実装

NPP の BSD/OS に実装について報告する [69][70]。カーネルは大きくわけてプロセス制御、ファイルシステム、通信機能、システム立ち上げの 4 つの部分から成るが、ネットワーク通信機能の部分の改造を行った。

ネットワーク通信機能の部分は図 3.37 のようになっている。機器割り込みによってデータ (イーサネットフレームなど) がネットワークインターフェースで受信され、入力メッセージ待ち行列を通じて通信プロトコル IP に渡される。ソースコードを解析した結果、入力待ち行列は図 3.38 のように、*ifqueue* 構造体にマッピングされている。IP は *IF_DEQUEUE* という関数によって待ち行列からパケットを取り出すが、その順番はもちろん FIFO である。パケットは待ち行列に入った順番で待ち行列から出ていく。

そこで、*ifqueue* 構造体を図 3.39 のように変更した。*MakeTable()* 関数によって、付加的な *table* 構造体を作成される。*precedence* は *TYPEOFSERVICE* フィールドの *PRECEDENCE* の値に相当する。*tccflag* はそのパケットにタイムウィンドウ指定があるかどうかをしめすフラグである。*discard* は、DISCARD オプションがセットされているかを示す。*D* は、3.4.3 で定義した *D* の値に対応している。*forward, backward* はそれぞれ、*table* 構造体の順方向、逆方向のポインタで、*location* は実際のパケットを差し示している。

パケットスケジューリングは、*table* 構造体を参照して、3.4.3 で定義したアルゴリズムに従ってプロトコル処理をするパケットを選ぶ。パケットスケジューリングは各パケットごとに行われる処理であり、実装は注意深く行う必要がある。できるだけ計算量を減らすように、テーブルを優先度ごとに区域分けしたり、時間制約の厳しさの度合を示す計算 D/T を展開して、固定部分と時間とともに変化する部分に分けて計算するような工夫を行った。

図 3.40 のように 2 台のノート PC、1 台のデスクトップ PC を利用してローカルネットワー

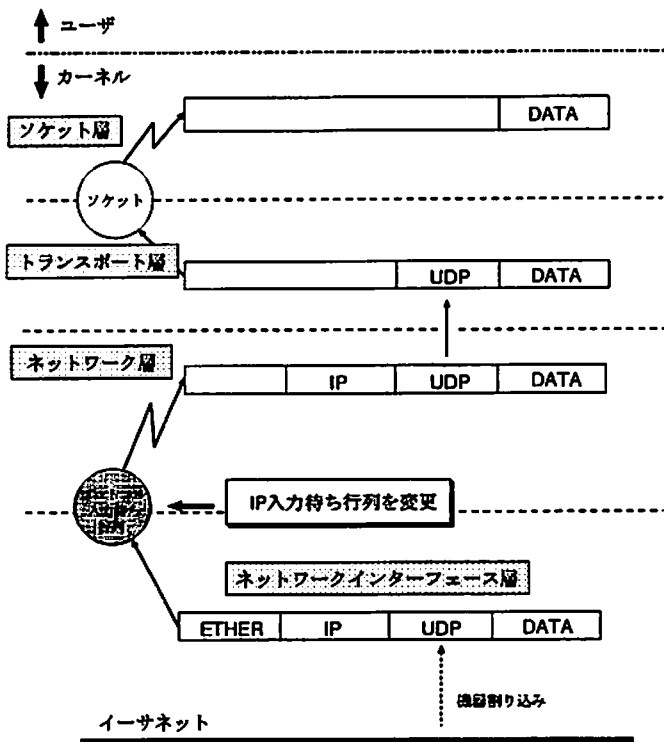


図 3.37: 内部構造

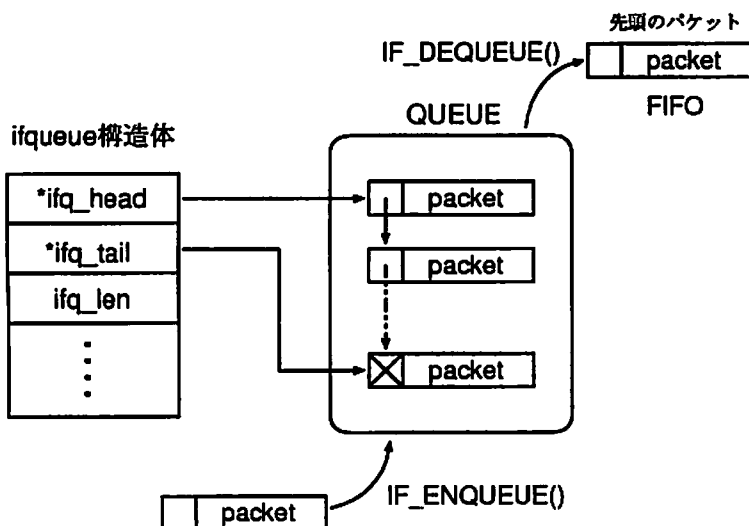


図 3.38: IP 入力待ち行列

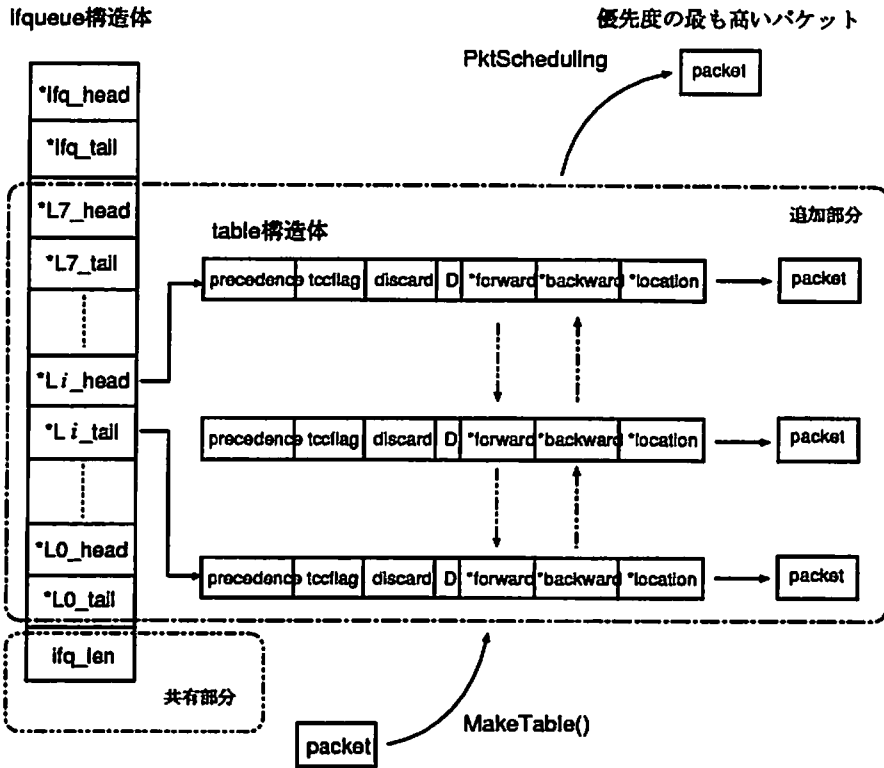


図 3.39: NPP 入力待ち行列

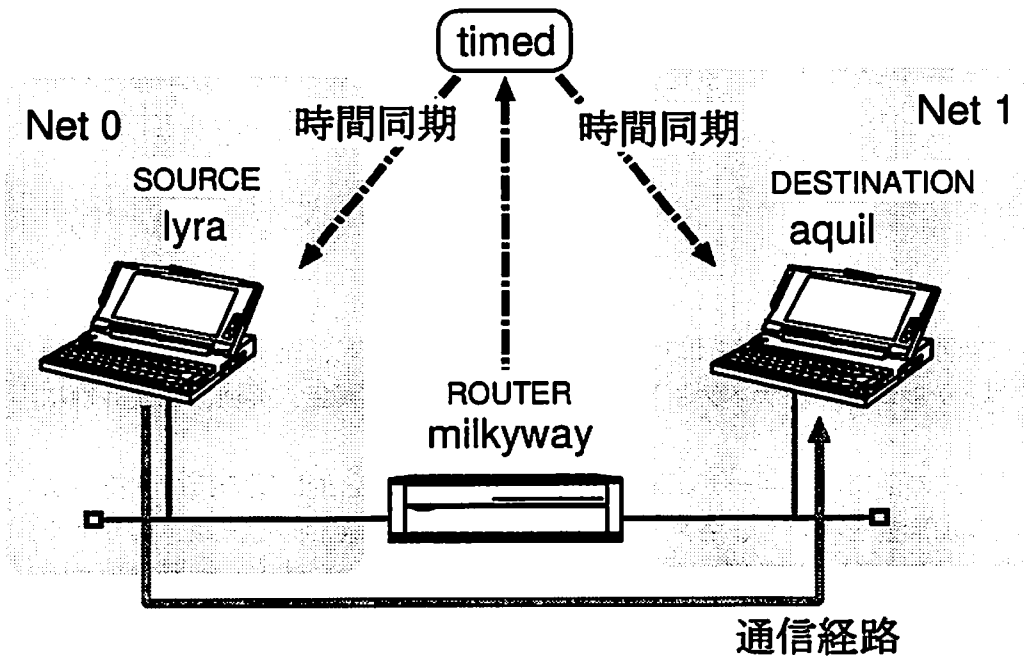
クを作り、実装した NPP の動作確認を行った。各 PC には NPP が組み込まれた BSD/OS 2.0 がインストールされている。lyra と aquil の 2 点間でタイムクリティカル通信を行い機能の動作テストを行った。設定した送信レートで連続的にパケットを送信する簡易なアプリケーションを作成し、lyra から aquil へと送信を行った。アプリケーションは UDP を用いてパケットを送信する。タイムウィンドウの設定は、IP オプションの設定を行う setsockopt() 関数を利用している。ルータ milkyway を介して lyra と aquil が通信できることを確かめた。実験、性能評価は今後の課題とする。

3.4.7 あとがき

本節ではタイムウィンドウ要求を持つデータ転送に対して、インターネット上でベストエフォートの実時間通信サービスを提供する実時間指向優先転送方式を提案し、そのネットワークプロトコルであるネットワーク応答特性プロトコル(NPP)について述べた。NPP はパケットの優先順位と時間制約の厳しさによって送受信処理するパケットを選択するパケットスケジューリングの機構を持つ。

また、あらかじめネットワークがサービス可能なタイムウィンドウを見積もり、フラグメント回避、バーチャルサーキットの確立などを行うネットワーク応答特性獲得機構と、ネットワーク応答特性モニタリングプロトコル NPMP について論じた。さらに、NPP をマルチメディア通信システムに対し適用するケースを考察した。NPP の性能に関してはシミュ

通信モデル



- 各計算機はtimedで時刻合わせを行う。この場合、milkywayに他の計算機は時刻を合わせる
- RouterにはNPPを実装。

図 3.40: 動作確認環境

レーションにより解析を行い、低負荷時には実時間通信を達成できること、高負荷時には低優先度のパケットの実時間通信は達成できないかわりに、高優先度のパケットの実時間制約が守られることを確かめた。また、現在のNPPの実装状況についても紹介した。

本節では保証を必要としない(資源の予約を必要としない)アプリケーションのための転送サービスを考察したが、一方で保証型のサービスを必要とするアプリケーションもある。ネットワークは最善努力型と保証型サービスの両方を提供すべきであり、本方式はその最善努力型のサービスの構築に貢献できるものとする。

今後の課題は、まず、実際の環境でのNPPの有効性の評価がある。実際に流れているトラフィックの種類とその割合を調べ、そのデータに基づきシミュレーション、実験を行いたい。次に、優先度NPPのマルチキャストのサポートに関する問題が挙げられる。遅延の異なる複数の終点に対してマルチキャストのパケットを発行する場合、どのようにタイムウィンドウを指定すべきは分かっていない。また、信頼性の保証やタイムクリティカル通信を要求するアプリケーションとのインタフェースを提供するNPPの上位プロトコルTPPの検討もする必要がある。

3.5 あとがき

本章では、同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式という4つのマルチメディア通信方式を提案し、実装、実験、性能評価について述べた。

同期・非同期通信方式の節では、ユーザが発行する参加形態要求と通信路の接続状態により、ユーザの会議の参加形態を決定し、各ユーザの参加形態から会議形態を決定、切り替える手順を述べた。また、集中型、分散型の会議情報の蓄積方法について述べ、切断中のユーザへの送信は分散型の蓄積により、実現できることを説明した。会議に欠席したユーザは後から蓄積された会議情報を閲覧することで会議進行を知ることができ、ネットワークから切断する機会の多い、モバイルユーザにとって有効であることが分かった。

選択的マルチメディア通信方式と先読み方式の節では、コンテンツベースの優先度について述べ、選択的転送と先読みのための選択的スケジューリングアルゴリズムについて論じた。また、サーバとミドルウェア、マルチメディアアプリケーションの実装についても報告した。実験により、選択的マルチメディア通信方式は、低速回線においても重要な情報を落とすことなくユーザに提供することができ、先読みにより転送が途切れたとしても再生を続行できることがわかった。蓄積型のビデオコンテンツの取得に対し、両方式は有効であるといえる。

実時間指向優先転送方式の節では、パケットの優先度と実時間制約に基づくパケットスケジューリングのアルゴリズム、プロトコルの規定等について述べた。シミュレーションにより、低負荷時には実時間通信を達成できること、高負荷時には低優先度のパケットの実時間通信は達成できないかわりに、高優先度のパケットの実時間制約が守られることを確かめた。また、現在の実装状況についても紹介した。

次章で、これらのマルチメディア通信方式を実際のマルチメディアアプリケーション、マルチメディア通信システムでどのように利用できるか、また、組み合わせて利用した場合の効果や問題点について述べる。また、マルチメディア通信方式の開発における開発ツールについても論じる。

第 4 章

モバイル電子会議システムへの適用

4.1 まえがき

本章では3で述べたマルチメディア通信方式を実際のマルチメディア通信システム、マルチメディアアプリケーションへの適用について検討する。

適用例としてモバイル電子会議システムを考え、そのシステムにおける同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式の役割について述べる。さらに、各通信方式を組み合わせて利用した際の効果、問題点について考察する。

その後、マルチメディアアプリケーションを開発するためのツールとして、実時間プロトコル設計言語とそのシミュレータについて紹介する。実際にモバイル電子会議システムの会議形態の切り替えのロジックの記述、実時間指向優先転送方式の設計仕様を記述し、シミュレータにより動作確認、性能評価を行う。

4.2 モバイル電子会議システム

本節では提案したマルチメディア通信方式のモバイル電子会議システムへの適用について述べる。モバイル電子会議システムはビデオコンテンツ取得と共に、今後、モバイルコンピューティング環境での利用の増加が期待されるマルチメディアアプリケーションである。

まず、モバイル電子会議システムを構築する上での要件について論じる。モバイル電子会議システムはマルチメディア電子会議システムに対する要求に加え、無線マルチメディア通信、モバイルホストを含む多地点通信、切断中の操作、非同期会議、蓄積機能などの機能を提供することが求められる。

そして、提案しているマルチメディア通信方式がそれらの要求のいくつかを解決できることを示す。同期・非同期通信方式は同期・非同期会議の会議形態の切り替え、会議情報の蓄積の機能を提供する。選択的マルチメディア通信方式と先読み方式は無線を含むインターネット上の蓄積型マルチメディア通信を、実時間指向優先転送方式は、ライブ型、蓄積型両方のマルチメディア通信をサポートする。

さらに、各マルチメディア通信方式を組み合わせたときの効果や問題点に関して考察する。

4.2.1 要件

以下でマルチメディア電子会議システム、モバイル電子会議システムのシステム要件を明らかにし、それに関する関連研究についてふれる。

マルチメディア電子会議システムの要件

マルチメディア電子会議システムには以下の機能が要求される。

- R1 ビデオ、ボイス機能：キャプチャ、圧縮、実時間（マルチキャスト）転送、同期（リップシンク）再生
- R2 データ会議機能（ホワイトボード、アプリケーション共有）：トークン処理、操作の順序保証を行う信頼性のある（マルチキャスト）転送、全参加者の共有情報の一貫性保証。
- R3 マルチメディア同期：R1 と R2 の同期、例えばホワイトボードの書き込みとボイスの同期など [71]。
- R4 多地点通信機能：MCU(Multipoint Control Unit) を利用する集中型、各端末が複数の宛先に配送する分散型、LAN 型の 3 種類の実現方法がある [72]。
- R5 相互接続性: データ会議の国際標準として ITU-T T.120 やビデオ会議標準の H.320 シリーズがある。

他にもセキュリティの機能や、会議を始める際のアナウンスや参加者を呼び出す機能なども必要になる。

モバイル電子会議システムの要件

- R6 無線マルチメディア通信：有線に比べ伝送速度が遅く、誤り率が高く、スループット、誤り率などが頻繁に変化するワイヤレス通信上でマルチメディア情報を配送すること。伝送速度の問題に関しては、ITU-T 勧告の低ビットレートのビデオ会議用映像符号化方式 H.263 や、ボイス符号化方式 G.729 などを利用できる。通信品質の変動に適應するビデオ会議システムの例として、H.261 符号化方式のパラメータを調節して出力データ量を増減したり [21]、マルチエージェントを利用してユーザ要求や計算機資源の状況に応じて QoS を自律的に調整するものがある [73]。
- R7 モバイルホストを含む多地点通信機能: インターネットにおけるモバイルホストを含むマルチキャストの議論がある [74]。

- R8 切断中の操作、切断からの回復：ネットワークから切断中にも発言、操作を行えること。例えば、切断中でもキャッシングによりファイルアクセス可能なファイルシステム [34] を利用できる。また、接続時に複数の書き込みによる衝突を検出、解決すること [35]、送受信中にユーザが自発的に切断したり、意図しない切断が起きても再接続時に途中から転送を開始するようなモバイルミドルウェア機能のサポート [31] 等も必要である。
- R9 非同期会議、蓄積機能：会議から切断する可能性のあるモバイルユーザのために、同期会議だけでなく、不在期間の会議情報の蓄積機能や非同期会議機能をサポートすること。
- R10 スムーズな途中参加、復帰：参加者の不在期間の会議の発言情報を効率的に提供し、会議からの遅れを迅速に取り戻せること。電子会議システムは、ビデオ、ボイス、ホワイトボードの書き込みをデジタルデータとしてストレージに蓄積する際に、誰がいつその発言、書き込みを行ったかを記録しておくことで、会議進行を保存できる。この会議進行情報をインデックスとして利用して、蓄積された会議発言情報に効率的にアクセスできると考える。類似研究として、ビデオ会議システムの模様を記録したビデオに対し、音声認識と辞書チェーン (lexical chaining) に基づいてインデックスをつけるシステム Jabber [75] がある。このシステムは、会議発言情報を議事録として利用したり、ノートとして利用することを考えている。

4.2.2 マルチメディア通信方式の役割

4.2.1に対し、本論文で提案した各マルチメディア通信方式の役割について述べる (図 4.1)。同期・非同期通信方式は R9 の非同期会議、蓄積機能の要求に対応する。3.2で論じたように、同期・非同期通信方式は、同期・非同期・部分非同期会議という3つの会議形態をサポートし、会議情報の蓄積を行うことができる。3.2.6で同期・非同期通信方式をビデオ電子会議システムに組み込み、実装を行った。実際に開発したビデオ電子会議システムを使って会議を行った結果、会議情報の蓄積は会議の欠席者に対して、有用だとの評価を得た。

選択的マルチメディア通信方式、先読み方式は R1 のビデオ、ボイス機能の実時間転送の要求、R6 の無線マルチメディア通信の要求に対応する。蓄積された会議情報の取得や、会議で利用する資料ビデオの取得に利用することができる。非同期会議、部分非同期会議における蓄積された会議情報をビデオ会議サーバ (あるいはホームサーバ) から取得する。また、会議で資料ビデオを皆で見ながら議論を行う場合、ビデオ会議サーバ、あるいはビデオコンテンツサーバから資料ビデオを取得する。選択的マルチメディア通信方式、先読み方式を利用することにより、低速で不安定な通信環境でも、安定した効率的なビデオ配送をユーザに提供することができる。

実時間指向優先転送方式も R1 のビデオ、ボイス機能の実時間転送の要求、R6 の無線マルチメディア通信の要求に対応する。蓄積型のマルチメディア通信に加え、ライブ型のマルチメディア通信、会議の制御用通信も扱うことができる。無線通信を含むインターネットと

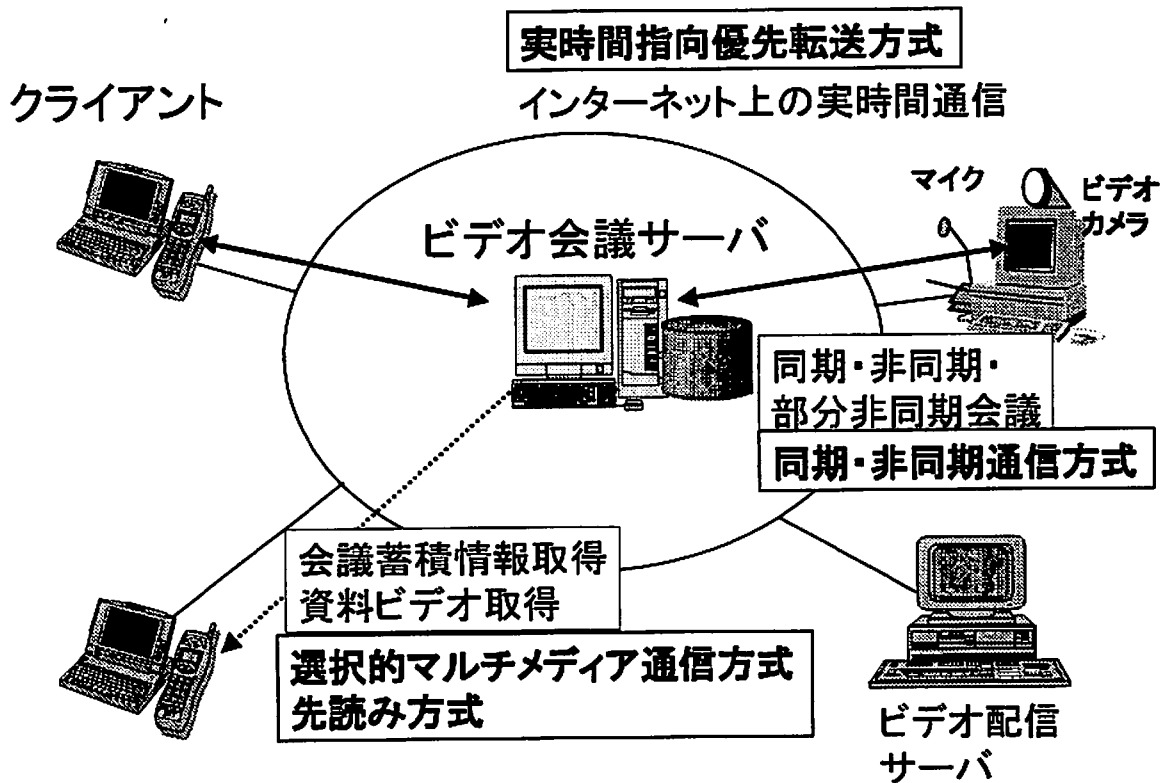


図 4.1: モバイル電子会議システム

いう低速、不安定で通信サービス品質が保証されない通信環境において、実時間性を要求するパケット、重要なパケットを優先的に転送する機構を提供できる。

図 4.2のように、同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式は、クライアント、サーバアプリケーションの一部、あるいはミドルウェアとして組み込まれる。一方、実時間指向優先転送方式は、クライアント、サーバとクライアント・サーバ間の経路上の各ノードにも組み込まれる。次節で、各マルチメディア通信方式の組み合わせに関して考察する。

マルチメディア通信方式の組み合わせ

同期・非同期通信方式との組み合わせ

- 選択的マルチメディア通信方式：同期・非同期通信方式における蓄積された会議情報の取得において、選択的マルチメディア通信方式を利用することができる。ただし、選択的マルチメディア通信方式はあらかじめ蓄積情報に人間が優先度を与えることを前提としている。会議の蓄積情報に誰かがこの発言は重要で、この発言は重要でないという優先度を与えればよいが、実用的ではないと思われる。

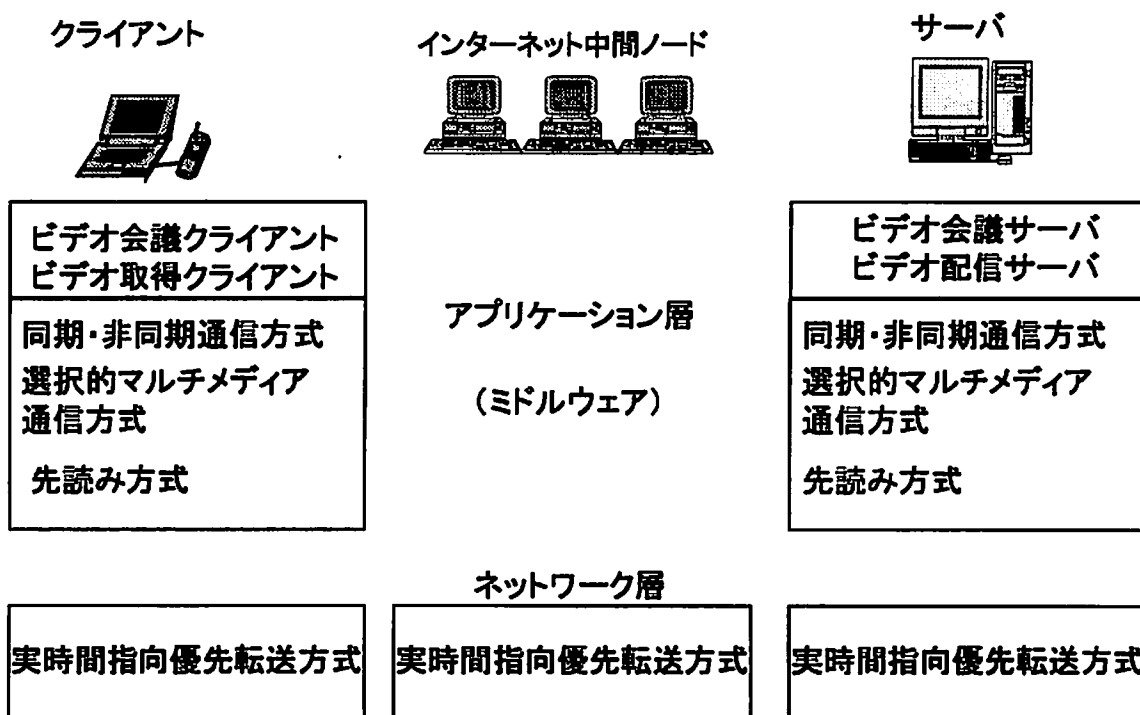


図 4.2: モーバイル電子会議システム

- **先読み方式:** 先読み方式は選択的マルチメディア通信方式のように優先度の付与を必要としない。蓄積された会議情報の取得に先読み方式を利用できる。
- **実時間指向優先転送方式:** 同期会議、部分非同期会議、非同期会議におけるマルチメディア情報の実時間通信をサポートする。非同期会議において発言を送信する場合のみ、実時間通信の必要はない。

選択的マルチメディア通信方式との組み合わせ

- **先読み方式:** 高優先度のビデオフレーム、オーディオブロックを選択的に先読みすることで、ユーザに提供するマルチメディア再生をより洗練させることができる。仮に転送が途切れても重要な情報を落とさず再生できる。
- **実時間指向優先転送方式:** 選択的マルチメディア通信方式の優先度を実時間指向優先転送方式の packets に与える優先度に反映させることで、輻輳、高負荷時に重要な情報を含む packets が落ちたり、遅れるのを防ぐことができる。

先読み方式との組み合わせ

	同期・非同期	選択的	先読み	実時間
同期・非同期	—	×	○	○
選択的		—	○	○
先読み			—	○
実時間				—

表 4.1: マルチメディア通信方式の組み合わせの効果

- 実時間指向優先転送方式：先読み情報は受信者に届くまでに時間的余裕があるので、パケットに与える実時間制約を比較的大きくできる。これにより、本当に実時間制約の厳しいパケットを優先して送ることができる。

4.3 マルチメディア通信システム開発ツール

ビデオ、オーディオなどのマルチメディア情報は時系列的に変化する連続データであり、ビデオは定期的な間隔で連続するビデオフレーム、オーディオは定期的な間隔で連続するオーディオサンプルから構成される。ビデオ、オーディオを通信でやりとりする場合、受信側は元々のビデオ、オーディオのタイミングを送信側でのタイミング再生する必要がある。すなわち、あるビデオフレーム、オーディオサンプルは受信側の再生タイミングまでに通信により届けられなければならない。つまり、ビデオフレーム、オーディオデータは実時間制約を持っている。

また、マルチメディア通信においては、メディア間同期とメディア内同期と呼ばれる同期の問題も必要である。メディア間同期には、ビデオ会議においてビデオの口の動きとボイスを一致させるリップシンクや、あるユーザがしゃべりながらホワイトボードに書き込みを行ったり、ポインタを動かしたときに、他のユーザの端末におけるボイスとホワイトボード、ポインタの動きの再生が同じタイミングで行われるようなマルチメディア同期がある。メディア内同期とは、元のマルチメディア情報のタイミングを保持して受信側で再生を行うものである。すなわち、再生の実時間に対する同期であり、30秒のビデオは早すぎることなく、遅れることなく30秒で再生する。

分散マルチメディア通信システムは制御情報をやりとりするが、制御情報の遅れは性能の劣化だけでなく障害にもなりうる。制御情報通信は低遅延性、ある上限以内の転送遅延を要求する。

このような実時間性をもつ通信システム、プロトコルの設計は複雑で、誤りや性能上のボトルネックとなる問題が入り込む可能性がある。従来、効率的であいまいでない通信プロトコルの設計のために形式記述技法 (FDT)[76]、が開発され、LOTOS[77]、Estelle[78]、SDL[79]、ペトリネットなどの記述言語やそのシミュレータ、バリデータ (検証器) が利用されてきた [80]。しかし、従来の FDT は実時間というよりも時間順序を主体としたものであり、マルチメディア通信、タイムクリティカル通信 [81] などの実時間の要求を含んだ通

信サービスの記述の能力は不十分であった。そこで、LOTOS-Tなどの従来の形式記述言語に実時間の概念をとり入れたものが提案された [82] [83][84][85]。

筆者らもプロトコル記述言語の一つである *PROMELA* [86] を時間特性について構文などの言語仕様を拡張し、時間概念を持った通信システム、プロトコルの仕様記述を可能にする *T-PROMELA* [87][88][89] を提案してきた。さらに、その仕様記述の動作のトレースや検査を可能とするシミュレータを開発した。

本節ではマルチメディア通信システムの開発ツールとして、実時間プロトコル設計言語 *T-PROMELA* の構文とそのシミュレータ *T-SPIN* の構造について紹介する。また適用例として、同期・非同期通信方式の会議形態の切り替え、実時間指向優先転送方式の仕様を *T-PROMELA* で記述し、*T-SPIN* でシミュレーションし、その動作確認、性能評価を行う。

4.3.1 実時間プロトコル設計言語

構文拡張

本論文では *T-PROMELA* の実時間通信に関する記述の構文とその記述例について触れる。その他の構文や詳細については、[87][88][89] を参照。

- 文実行時間記述

明示的な実行時間の記述

- 構文 シーケンス `sequence ::= step [t_exprs]`
`{ ';' step [t_exprs] }`

ステップ `step ::= [decl_lst] stmt`

文の後に実行時間の式を記述してもよい。(なくてもよい)

実行時間の式は以下のように定義される。

実行時間記述 `t_exprs ::= COST t_expr ['::' t_expr]`

実行時間の式 `t_expr ::= '(t_expr)'`

| `t_expr binop t_expr`

| `var_ref`

| `DCONST`

| `CONST`

`COST`(これは実際は''@''を意味する)の後に、数や変数、式などを記述する。記号::をはさんで実行時間の範囲を指定することもできる。

- 例 `channel ! data @10`

チャンネルに `data` を送信するのに 10ms かかる。

`in ? ack @20 :: 40`

チャンネル in から ack を入力するのに 20 から 40ms かかる。

- 伝送時間記述

明示的な伝送時間の記述

- ・ 構文 チャンネル宣言

```
ch_init ::= '[' CONST ']' OF
          '{' TYPE { ',' TYPE }* '}' [ t_exprs]
```

チャンネル宣言時にそのチャンネルの伝送時間を記述できる。伝送時間記述は上の実行時間と同じである。

- ・ 例 `chan ipc_channel = [2] of { byte, byte } @20`
チャンネル `ipc_channel` の伝送時間は 20(ms) である。

- 時間型の変数

実数型で、時間を記述するための変数

- ・ 構文 宣言 `one_decl ::= [TYPE ivar { ',' ivar } *]`
宣言リスト `decl_lst ::= one_decl { ';' one_decl } *`
初期化変数 `ivar ::= var_dcl | var_dcl ASGN expr`
`| var dcl ASGN ch_init`

- ・ 例 `time timeout = 5.0`
時間型変数 `timeout` を初期値 5.0 として宣言する。

- 時間変数 `gtime`

シミュレーション開始からの時間経過を示すシステム予約型の変数

- ・ 構文 `time gtime`
`time` 型の変数と同様に使用できるが代入はできない(初期化はできる)。これはシステムにより自動的に更新される。

- ・ 例 `cur_time = gtime`
現在の経過時間を時間型変数 `cur_time` に代入する。

- リソースの宣言

```
rstype: RSTYPE ASGN '{' args '}'
      | ;
```

例: `rstype = { aries, taurus, leo } ;`
リソース (CPU) として、`aries`、`taurus`、`leo` を宣言する。

- プロセスが生成されるリソースの記述

```
expr: RUN NAME '(' args ')' IN expr
```

例: `run process1() in aries`
`aries` というリソース上でプロセス 1 を生成する。

- 各マシンごとのローカルタイムの記述

```
time ltime ;
```

```
例: proctype process1()  
{  
  time ltime ;  
  cur_time = ltime
```

`process1` の局所的な時間を表す変数が `ltime` である。
他のマシンとの時間の一貫性を保つためには、`gtime` を用いる。

- 送信メッセージサイズの記述

```
stmt: varref SND margs ':' expr
```

例: `channel ! id,data:512`
チャンネル `channel` に `id,data` を送信するが、この送信で実際に送るバイト数は 512 バイトである。

これまでは、メッセージのサイズが時間測定に反映されていなかったが、これからは送信時に実際に記述されたバイト数の送信時間、伝送時間を測定する。また受信時は、メッセージに付属してくるメッセージサイズに従って受信時間を測定する。

- 非決定的選択処理の重み付け

```
option : SEP '[' expr ']'  
        sequence
```

例:

```

do
:: in ? data ->
    if
        :: [80] out ! data
        :: [20] out ! error
    fi
od

```

例えば信頼性の低い通信路を記述する場合、上のようになる。
ここでは、20%の確率でメッセージを歪める。

記述例

構文拡張の応用により、次のような記述が可能になる。

- タイムアウトなどの明示的な記述

例	<code>chout ! data ;</code>	送信
	<code>st_time = gtime ;</code>	タイマスタート時の時間保存
	<code>do</code>	
	<code>:: chin ? ack</code>	確認応答受信
	<code>:: (gtime - st_time > 5000)-></code>	タイムアウト監視
	<code>printf('timeout occurs')</code>	タイムアウト発生
	<code>od</code>	

チャンネル `chout` に `data` を送信した後、チャンネル `chin` より `ack` を受信するが、そのタイムアウトを5秒(5000ms)に設定する。タイムアウト監視は、(現在の時間 - タイマスタート時の時間)と5000の比較により行われる。

- 表明文による「ある実行列がある時間以内に完了しなければならない」というような実時間要求の記述

例	<code>st_time = gtime ;</code>	タイマスタート時の時間保存
	<code>chout ! req ;</code>	トランザクション処理
	<code>chin ? ack ;</code>	トランザクション完了
	<code>assert(gtime-st_time<2000) ;</code>	トランザクションが2s以内かの表明

トランザクションが2000(ms)=2(s)以内に終了しなかった場合、表明違反としてユーザに通知する。

4.3.2 実時間プロトコルシミュレータ

*T-SPIN*は *T-PROMELA* の仕様記述をシミュレーションするツールである。*PROMELA* の *SPIN* シミュレータを *T-PROMELA* の構文を扱えるように拡張し、従来は文の時間順序の関係で動作していたのをクロックベースで動作するように修正した。

機能

T-SPIN シミュレータには *SPIN* の機能から受け継いだものと新たに加えられたものがある。

SPIN から継いだ機能として以下がある。

- 大域変数、局所変数の値の表示 (チャンネルの内部状態も含む)
- 空きのないキューへ送信した場合、そのメッセージを失うか、または送信不可能とするかの処理の選択
- 乱数のシードの選択 (シミュレーション実行の再現を可能にする)
- 送受信イベントの表示
- 1文ごとのトレース
- 表明文による、仕様に対する要求の検査

これらが *T-SPIN* の新しい機能である。

- ラウンドロビン・スケジューリング方式のクオンタム設定
- 時間経過の逐次表示
- 表明文による仕様に対する実時間要求の検査

クオンタムとはプロセスに対して CPU を割当てる時間のことで、あるプロセスはクオンタムの時間だけ実行され、その時間内に終了しない場合ディスパッチャによってプロセス実行可能列の最後に移される。このクオンタムはマシン、システムによって異なるので、ユーザは任意の時間に設定できる (デフォルトは 100(ms))。時間経過の逐次表示は、シミュレーション開始よりの経過時間を、ユーザが設定した時間間隔で表示するものである。これらの機能により、動作の確認や実行時間の評価ができる。

構造

T-SPIN は以下の事項を仮定している。

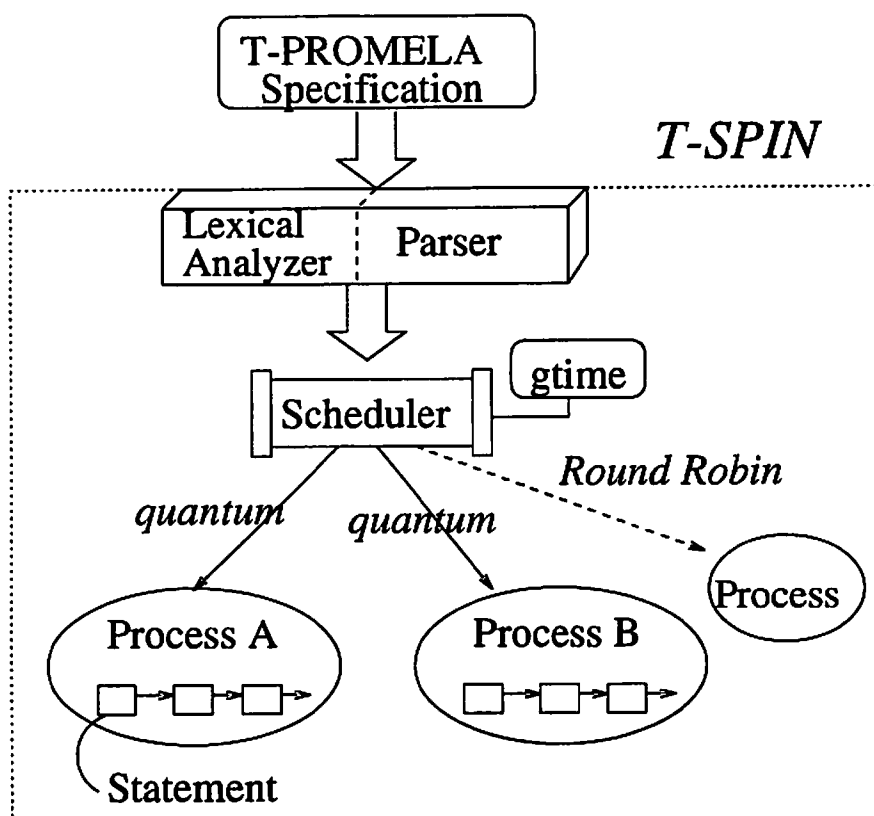


図 4.3: T-SPIN 構造

1. ラウンドロビン方式

T-PROMELA は並行プロセスを扱うが、プロセスのスケジューリングはラウンドロビン方式で行う。例えば UNIX では CPU 時間を累積すればするほど、優先度を下げることを行っている。一方、待ち時間に対してはスタベーションを防ぐため、エージング機構をとりいれていて、ラウンドロビンのクオンタム (0.1 秒程度) を消費するごとに優先度を計算するという処理を行っている。これに対し、*T-SPIN* は簡単のためラウンドロビン方式を用いた。

2. 実行時間の内部計測

各文に実行時間が記述してあった場合はその値を実行時間とし、記述のない場合はシミュレータが実際の実行時間を計測して、その値を用いる。あらかじめ実行時間がわかっている文に対しては、その実行時間記述に従ってスケジューラの時間の更新を行うが、そうでない場合はシミュレータ自身がその文を実行する際タイマを起動して、実行時間を計測する。

T-SPIN の構造を図 4.3.2 に示す。

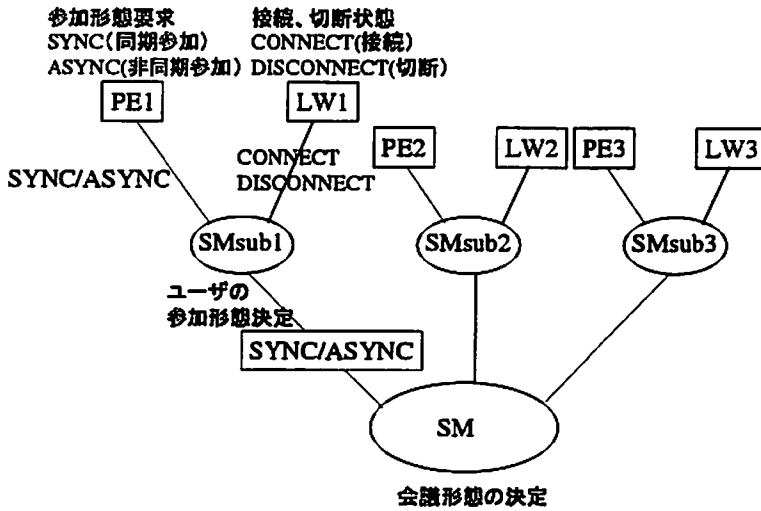


図 4.4: 対象モデル (同期・非同期通信方式)

4.3.3 同期・非同期通信方式への適用

T-PROMELA の構文拡張とその応用を説明したところで、実際の仕様記述を行うことにする。3.2で述べた同期・非同期通信方式の会議形態の切り替えの仕様を記述し、シミュレーションにより動作確認する。

対象モデルと記述

以下に、*T-PROMELA* によって記述した仕様を示し、図 4.4にこの通信システムを示す。参加者エンティティ PE が参加形態要求を、通信路ウォッチャ LW が接続/切断状態を、セッションマネージャ SM に通知し、SM が各参加者の参加形態を決定した後、会議形態を決定する。

```

/*****
synchronous /asynchronous human communication channel
Spec
*****/
rstype = { Client1, Client2, Client3, Server };

/* チャンネル */
chan pe1sm1 = [1] of { int }; /* PE -> SM */
chan lw1sm1 = [1] of { int }; /* LW -> SM */
chan pe2sm2 = [1] of { int }; /* PE -> SM */
chan lw2sm2 = [1] of { int }; /* LW -> SM */
chan pe3sm3 = [1] of { int }; /* PE -> SM */
chan lw3sm3 = [1] of { int }; /* LW -> SM */
chan smsub2sm = [1] of { int, int }; /* SM内の通信 */

```

```

#define SYNC          1 /* 同期参加 */
#define ASYNC        0 /* 非同期参加 */
#define CONNECT      1 /* 接続状態 */
#define DISCONNECT   0 /* 非接続状態 */
#define SYNC_CONF    0 /* 同期会議 */
#define ASYNC_CONF   1 /* 非同期会議 */
#define PARTIALASYNC_CONF 2 /* 部分非同期会議 */

int m=0; /* 全参加者数 */
int n=0; /* 同期参加の人数 */

proctype PE(chan pe2sm) /* 参加者エンティティ */
{
    time ltime;
    int current = ASYNC; /* 現在の参加形態 */
    do
    :: (current == ASYNC)->
        pe2sm ! SYNC; /* 同期参加 */
    :: (current == SYNC)->
        pe2sm ! ASYNC; /* 非同期参加 */
    od
}

proctype LW(chan lw2sm) /* 通信路ウォッチャ */
{
    time ltime;
    int current = DISCONNECT; /* 現在の参加形態 */
    do
    :: (current == DISCONNECT)->
        lw2sm ! pid, SYNC /* 接続状態 */
    :: (current == CONNECT)->
        lw2sm ! pid, ASYNC /* 切断状態 */
    od
}

/* セッションマネージャ */
proctype SHsub(int pid, chan pe2sm, lw2sm)
{
    time ltime;
    int pe=ASYNC;
    int lw=DISCONNECT;
    do
    :: if
        :: pe2sm ? pe
        :: lw2sm ? lw
        fi ;
        if
        :: (pe == SYNC && lw==CONNECT) ->
            smsub2sm ! pid, SYNC;
        :: (pe == ASYNC || lw==DISCONNECT) ->
            smsub2sm ! pid, ASYNC;
        fi
    od
}

```

```

proctype SM() /* セッションマネージャ */
{
    /* 会議形態決定 */
    time ltime;
    int user1=ASYNC, user2=ASYNC, user3=ASYNC;
    int form = ASYNC_CONF;
    do
    :: if
        :: smsub2sm ? 1, user1
        :: smsub2sm ? 2, user2
        :: smsub2sm ? 3, user3
    fi;
    n = user1 + user2 + user3;
    if
    :: (n==m) ->
        form = SYNC_CONF /* 同期会議 */
    :: (n > 1 && n < m) ->
        form = ASYNC_CONF /* 非同期会議 */
    :: (n <= 1) ->
        form = PARTIALASYNC_CONF /* 部分非同期会議 */
    fi
    od
}

init /* 初期化プロセス */
{
    run PE(pe1sm1) in Client1;
    run LW(lw1sm1) in Server;
    run SMsub(1, pe1sm1, lw1sm1) in Server;

    run PE(pe2sm2) in Client2;
    run LW(lw2sm2) in Server;
    run SMsub(2, pe2sm2, lw2sm2) in Server;

    run PE(pe3sm3) in Client3;
    run LW(lw3sm3) in Server;
    run SMsub(3, pe3sm3, lw3sm3) in Server;

    m = 3; /* 参加者数 3 */
    run SM() in Server;
}

```

シミュレーション

この仕様の動作確認を行うため、*T-SPIN*でシミュレーションする。上記の仕様を *syncasync.tpr* というファイル名で保存し、以下のようなコマンドを実行する。*tspin*が *T-SPIN*に対応するプログラムである。*-r-s*はプロセス間の受信、送信を表示するオプションである。シミュレーションを行うと以下のようなレポートが得られる。

```

#tspin -r -s syncasync.tpr
...(中略)
非同期会議 ユーザ 1[0] ユーザ 2[0] ユーザ 3[1]
proc 3 (LW) line 45, Send 1.0 -> queue 2 (lw2sm)

```

```

Send time 0.0 transport 30.0
ユーザ1 接続状態
proc 4 (SMsub) line 65, Recv 1.0 <- queue 2 (lw2sm)
Receive at 128400.0 end 128400.0
proc 4 (SMsub) line 69, Send 1.0, 1.0 -> queue 3 (smsub2sm)
Send time 0.0 transport 0.0
proc 1 (SM) line 83, Recv 1.0, 1.0 <- queue 3 (smsub2sm)
Receive at 128500.0 end 128500.0
部分非同期会議 ユーザ1[1] ユーザ2[0] ユーザ3[1]
proc 6 (LW) line 50, Send 0.0 -> queue 5 (lw2sm)
Send time 0.0 transport 30.0
proc 3 (LW) line 50, Send 0.0 -> queue 2 (lw2sm)
Send time 0.0 transport 30.0
ユーザ1 切断状態
proc 4 (SMsub) line 65, Recv 0.0 <- queue 2 (lw2sm)
Receive at 130400.0 end 130400.0
proc 4 (SMsub) line 71, Send 1.0, 0.0 -> queue 3 (smsub2sm)
Send time 0.0 transport 0.0
proc 1 (SM) line 83, Recv 1.0, 0.0 <- queue 3 (smsub2sm)
Receive at 130500.0 end 130500.0
非同期会議 ユーザ1[0] ユーザ2[0] ユーザ3[1]

```

非同期会議状態からはじまり、LW から SMsub へユーザ1 が接続状態との通知がなされる。次に、SMsub から SM へユーザ1 が同期参加であるとの通知がなされ、部分非同期会議に会議形態が切り替わる様子が確認できる。その後、ユーザ1 が切断し、再び非同期会議に切り替わっている。

このように、マルチメディア通信システムの開発者は、通信プロトコルやロジックを *T-PROMELA* でシンプルに記述し、*T-SPIN* で通信手順、ロジックを確かめ、デバッグすることができる。仕様記述が正しいことが確認できたら、その記述を元に C 言語、C++ 言語で実装を行うことができる。*T-PROMELA* の記述は C 言語に似たものであり、実装への移行は難しくない。同期・非同期通信方式の通信プロトコルは比較的簡単なものではあるが、仕様を記述し、動作確認することで実際の実装をスムーズに行うことができた。複雑なプロトコルでは、この開発手順がより有効に働くと思われる。

4.3.4 実時間指向優先転送方式への適用

対象モデルと記述

次に、3.4で述べた実時間指向優先転送方式の記述を行う。以下に3.4.5でシミュレーションを行ったシステム記述を示す。ここではタイムウィンドウによるスケジューリングは行わず、パケットの優先度にも基づいたスケジューリングを示す。また、スタベーションの回避に関しては優先度を上げるのではなく、各優先度のキューに対して重み付けを行うことにより回避する。

```

/*****
Network Protocol with Performance    for T-SPIN ver 2.0
Simulation Spec

```

```

*****/

rstype = { CPU1, CPU2, CPU3, CPU4, CPU5, CPU6, CPU7, CPU8 };

#include "NPP.ptr"
/*
#define LOAD_MMD      40000-2000 マルチメディア負荷発生率
#define LOAD_NTC      0.2 or 0.4
*/

#define T_SERVICE      1000 /* サービス時間 */
#define T_REQUEST_TC   10000 /* TC メッセージ発生率 */
#define T_REQUEST_MMD  3333 /* マルチメディアデータ発生率 */

#define LOAD_TC        10000 /* TC メッセージ負荷 */
#define QUE_LEN        1024 /* 待ち行列 */

#define QSIZE          50 /* ルータのスロット数 */
#define TW_TC          7000 /* TC メッセージタイムウィンドウ */
#define TW_MMD         8000 /* マルチメディアデータ TW */
#define WAIT           skip

chan p2sTC = [ QUE_LEN ] of { time }; /* プロデューサ -> TC メッセージ */
chan p2sMMD = [ QUE_LEN ] of { time }; /* マルチメディア */

chan s2rTC = [ QUE_LEN ] of { int, time }@T_SERVICE; /* ソース -> ルータ */
chan s2rMMD = [ QUE_LEN ] of { int, time }@T_SERVICE;
chan s2rNTC = [ QUE_LEN ] of { int, time }@T_SERVICE;

chan r2dTC = [ QUE_LEN ] of { time }@T_SERVICE; /* ルータ -> 受信ホスト */
chan r2dMMD = [ QUE_LEN ] of { time }@T_SERVICE;

int A1, A2 ; /* TC メッセージ、マルチメディアデータの送信数 */
int L1, L2, L3 ; /* 負荷メッセージの送信数 */
int C1, C2 ; /* TC メッセージ、マルチメディアデータの受信数 */
int TC_LOSS, MMD_LOSS, NTC_LOSS ; /* メッセージの紛失数 */

time WTC = 0., WMMD = 0. ;
time T = 60000000. ;
int OK_TC = 0 , OK_MMD = 0 ;

time gtime = 0. ;
int LOSS = 0 ;

proctype procl() /* TC メッセージ生成 */
{
    time ltime;
    do
        :: ( gtime < T ) ->
A1 = A1 + 1 ;
        p2sTC ! ltime ;
        WAIT @ expntl( T_REQUEST_TC )
        :: ( gtime >= T ) -> break
    od
}

```

```

}

proctype proc2() /* マルチメディアデータ生成 */
{
    time ltime;
    do
        :: ( gtime < T ) ->
A2 = A2 + 1 ;
        p2sMMD ! ltime ;
        WAIT @ expntl( T_REQUEST_MMD )
        :: ( gtime >= T ) -> break
    od
}

proctype src() /* 送信ホスト */
{
    time ltime;
    time t2 ;
    do
        :: [2]( gtime < T && len(p2sTC) > 0 ) ->
        WAIT @ T_SERVICE ;
        p2sTC ? t2 ;
        s2rTC ! 0, t2
        :: [1]( gtime < T && len(p2sMMD) > 0 ) ->
        WAIT @ T_SERVICE ;
        p2sMMD ? t2 ;
        s2rMMD ! 1, t2
        :: ( gtime >= T ) -> break
    od
}

proctype load1() /* 負荷： TC メッセージ */
{
    time ltime;
    do
        :: ( gtime < T ) ->
L1 = L1 + 1 ;
s2rTC ! 3, ltime ;
        WAIT @ expntl( LOAD_TC )
        :: ( gtime >= T ) -> break
    od
}

proctype load2() /* 負荷：マルチメディアデータ */
{
    time ltime;
    do
        :: ( gtime < T ) ->
L2 = L2 + 1 ;
s2rMMD ! 3, ltime ;
WAIT @ expntl( LOAD_MMD )
        :: ( gtime >= T ) -> break
    od
}

```



```

proctype load3() /* 負荷： 通常データ */
{
    time ltime;
    do
        :: ( gtime < T ) ->
L3 = L3 + 1 ;
s2rNTC ! 3, ltime ;
WAIT @ expntl( LOAD_NTC )
        :: ( gtime >= T ) -> break
    od
}

proctype router() /* ルータ */
{
    time ltime;
    time t2 ;
    int n ;
    do
        :: ( gtime < T ) ->
if /* ルータ輻輳 & パケットロス */
:: ( len(s2rTC)+len(s2rMMD)+len(s2rNTC) > QSIZE ) ->
if
:: ( len(s2rNTC)>0 ) -> /* まず通常データロス */
s2rNTC ? n, t2 ;
NTC_LOSS = NTC_LOSS + 1
:: ( len(s2rNTC)==0 && len(s2rMMD)>0 ) ->
s2rMMD ? n, t2 ; /* 次にマルチメディアデータロス */
MMD_LOSS = MMD_LOSS + 1
:: ( len(s2rNTC)==0 && len(s2rMMD)== 0 ) ->
s2rTC ? n, t2 ;
TC_LOSS = TC_LOSS + 1
fi
:: ( len(s2rTC)+len(s2rMMD)+len(s2rNTC) <= QSIZE )
fi;
if
:: ( len(s2rTC) > 0 ) ->
if
:: [18]( len(s2rTC) > 0 ) -> /* TCデータの送受信 */
WAIT @ T_SERVICE ;
s2rTC ? n, t2 ;
if
:: ( n == 0 ) -> r2dTC ! t2
:: ( n != 0 )
fi
:: [3]( len(s2rMMD) > 0 ) -> /* マルチメディアデータ */
WAIT @ T_SERVICE ;
s2rMMD ? n, t2 ;
if
:: ( n == 1 ) -> r2dMMD ! t2
:: ( n != 1 )
fi
:: [1]( len(s2rNTC) > 0 ) -> /* 通常データ */
WAIT @ T_SERVICE ;

```

```

                s2rNTC ? n, t2
        fi
:: ( len(s2rTC) == 0 ) ->
if
    :: [3]( len(s2rMMD) > 0 ) -> /* マルチメディアデータ */
    WAIT @ T_SERVICE ;
    s2rMMD ? n, t2 ;
    if
        :: ( n == 1 ) -> r2dMMD ! t2
        :: ( n != 1 )
    fi
    :: [1]( len(s2rNTC) > 0 ) -> /* 通常データ */
    WAIT @ T_SERVICE ;
        s2rNTC ? n, t2
    :: [1]( len(s2rNTC) ==0 && len(s2rMMD)==0 )
    fi
fi ;
WAIT@1
    :: ( gtime >= T ) -> break
od
}

proctype dst() /* 受信ホスト */
{
    time ltime;
    time t2 ;
    do
        :: ( gtime < T ) ->
if
    :: [18]( len(r2dTC) > 0 ) -> /* TCデータの受信 */
C1 = C1 + 1 ;
    WAIT @ T_SERVICE ;
    r2dTC ? t2 ;
WTC = WTC + ltime - t2 ; /* (ltime-t2) パケット消費時間 */
    if
        :: ( ltime -t2 < TW_TC ) -> OK_TC = OK_TC + 1
        :: ( ltime -t2 >= TW_TC )
    fi
fi
:: [3]( len(r2dMMD) > 0 ) -> /* マルチメディアデータ */
    C2 = C2 + 1 ;
    WAIT @ T_SERVICE ;
    r2dMMD ? t2 ;
    WMMD = WMMD + ltime - t2 ;
if
:: ( ltime -t2 < TW_MMD ) -> OK_MMD = OK_MMD + 1
    :: ( ltime -t2 >= TW_MMD )
fi
:: [1]( len(r2dMMD) ==0 && len(r2dTC)==0 )
fi;
WAIT@1
    :: ( gtime >= T ) -> break
od ;
printf("TC Data: 到着率 スループット 平均システム時間 時間制約保証\n") ;

```

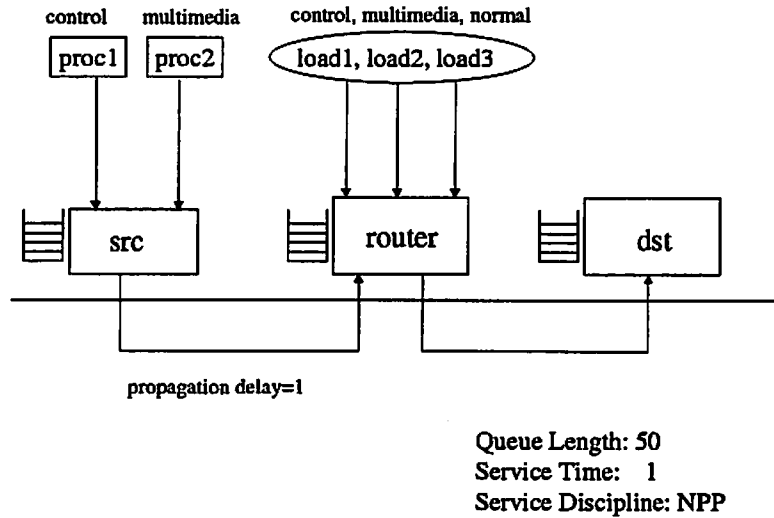


図 4.5: 対象モデル (実時間指向優先転送方式)

```
printf("%x %x %x %x\t", A1*1000/T, C1*1000/T, WTC/1000/C1, OK_TC / C1 ) ;
printf("%x %x %x %x\t", A2*1000/T, C2*1000/T, WHMD/1000/C2, OK_HMD / C2 ) ;
printf("%x %x %x %x %x %x \n", L1*1000/T, L2*1000/T,
      L3*1000/T, TC_LOSS/(L1+A1), MMD_LOSS/(L2+A2), NTC_LOSS/(L3) );
```

```
init /* 初期化プロセス */
{
  run proc1() in CPU1;
  run proc2() in CPU2;
  run router() in CPU3 ;
  run src() in CPU4;
  run dst() in CPU5 ;
  run load1() in CPU6;
  run load2() in CPU7;
  run load3() in CPU8
}
```

図 4.5にこのシステムを図示する。プロセス proc1, proc2 は、マルチメディアアプリケーションが生成するトラフィックである。proc1 は制御メッセージ、proc2 はマルチメディアデータに対応する。プロセス src, router, dst は始点、ルータ、終点に対応し、それぞれにネットワーク応答特性プロトコル NPP が実装されており、優先度に基づいたパケットスケジューリングを行う。

src は proc1, proc2 からトラフィックを受け取り、ルータに転送する。このとき src は優先度に基づいたパケットスケジューリングを行う。すなわち、制御メッセージのキューにメッセージがあったらそれを優先して送り、なかったらマルチメディアデータを送る。src 以外にもルータに負荷をかけるプロセス load1, load2, load3 が存在し、それぞれ制御メッセー

ジ、マルチメディアデータ、通常データを生成する。ルータはプロセス router に対応し、プロセス src や load1,2,3 からパケットを受け取り、制御メッセージ、マルチメディアデータ、通常データの順で処理する。src から来たパケットは終点 dst に転送する。dst は router からパケットを受け取り、優先度順で上位アプリケーションに渡す。このとき、パケットの実時間制約、すなわちタイムウィンドウが守られたかを検査する。

シミュレーション

この *T-PROMELA* による仕様記述を、*T-SPIN* でシミュレーションする。上記の仕様記述ファイルが *NPP.tpr* というファイル名で保存し、以下のコマンドを実行した。

```
#tspind NPP.tpr WLOAD2.pls
```

WLOAD2.pls は以下のようにシミュレーションを行う際のパラメータを保持するファイルである。ここでは2つのパラメータ *LOAD_MMD* と *LOAD_NTC* があり、それぞれマルチメディアデータ、通常データの負荷トラフィックを意味している。それぞれ20個の値が指定されているが、これはマルチメディアデータの負荷 *LOAD_MMD* を増加させ(ここでの数値は到着間隔を示しているので、到着間隔の減少は到着率の増加を意味する)、通常データの負荷 *LOAD_NTC* を一定にするというシナリオに相当する。

```
LOAD_MMD int
```

```
40000 20000 13333 10000 8000 6667 5714 5000 4444 4000 3636 3333  
3076 2857 2666 2500 2353 2222 2105 2000
```

```
LOAD_NTC int
```

```
5000 5000 5000 5000 5000 5000 5000 5000 5000 5000  
5000 5000 5000 5000 5000 5000 5000 5000 5000 5000
```

他のパラメータファイルを使って異なるシナリオでシステムの挙動を知ることができる。上記の記述例は *NPP* であるが、同様に *IP* の記述も行い、両方ともシミュレーションを行い、3.4.5で述べたような結果が得られた。*NPP* により負荷があがっても制御メッセージは低遅延かつ高い確率でタイムウィンドウ内に転送されることができ、マルチメディアデータも *IP* に比べ低遅延で配送されることができることが分かった。しかし、マルチメディアデータの高負荷時には、マルチメディアデータの遅延は指数的に増加し、時間制約を守ることができなくなるという問題点が明らかになった。

このように *T-PROMELA* で実時間システム、分散システム、マルチメディア通信システムを記述し、*T-SPIN* でシステムの性能を評価し、問題点を明らかにすることができる。

4.4 あとがき

本章ではまず、同期・非同期通信方式、選択的マルチメディア通信方式、先読み方式、実時間指向優先転送方式の適用例としてモバイル電子会議システムを考え、各マルチメディア通信方式をどのように利用できるか、そして、各通信方式を組み合わせた場合にどのような効果、問題点があるかについて論じた。

同期・非同期通信方式は同期・非同期・部分非同期会議という自由度の高いヒューマンコミュニケーションをサポートすることができ、先読み方式は蓄積された会議情報の取得や、会議で利用する資料ビデオなどの蓄積コンテンツの取得に利用できる。選択的マルチメディア通信方式も蓄積コンテンツの取得に有用である。実時間指向優先転送方式はライブ型、蓄積型両方のマルチメディア通信で利用できる。マルチメディア通信方式の組み合わせについては、同期・非同期通信方式と選択的マルチメディア通信方式の組み合わせは実用的ではないものの、その他の組み合わせには効果があることが分かった。

次に、実時間通信プロトコル設計言語を使って同期・非同期通信方式、実時間指向優先転送方式の記述を行い、シミュレータで動作確認、性能評価実験を行った。仕様を自然言語で記述し、プログラムを開発し、その都度試験するという通常の開発手順に比べ、プロトコル設計言語でシンプルなロジックを記述し、動作させることで、早い段階で仕様の誤りを見つけたり、性能の予測をたてることが可能になる。マルチメディア通信アプリケーションのように実時間の概念を持ち、複雑で、かつ性能を要求するようなシステムの設計において、実時間プロトコル設計言語とシミュレータの利用は効果があることが分かった。

第 5 章

結論

本論文は移動中や出先から必要なときにマルチメディア情報の発信、受信を行うことを可能にするマルチメディア通信方式について論じた。インターネットと広域無線通信が連携したネットワーク環境においてマルチメディア通信を行う際、無線が低速であること、無線とインターネット両方でスループットや遅延などが変動し、不安定であるという通信サービス品質の課題と、ユーザがインターネット上を移動し、インターネットから頻繁に切断するといったモバイルコンピューティングの課題があることを述べ、これに対し4つのマルチメディア通信方式を提案し、その検討、設計、実装、評価を行った。以下で、それぞれの通信方式に関して、得られた知見について述べる。

同期・非同期通信方式に関して、まず、非同期会議、部分非同期会議における会議情報の蓄積は、ネットワークから切断する機会の多いモバイルユーザにとって有効であることが分かった。会議から一時的に欠席しても後から蓄積情報を閲覧して不在期間の会議内容を知ることができる。また、蓄積された会議情報のインデックスとして、誰がいつ発言したかを示すグラフ表示機能も会議進行を理解する上で有用であった。40分の会議における蓄積されたボイス容量を調べた結果、蓄積に必要とするストレージ容量はそれほど多くなく、現在の一般的なハードディスクで十分であることが分かった。ユーザの接続、切断状態を監視するために定期的にユーザ端末が会議サーバにメッセージを送る機構を提案、実装し、動作確認を行った結果、低速な通信環境ではタイムアウトの設定に注意が必要であることがわかった。

選択的マルチメディア通信方式については、実装、実験を行った結果、重要なシーンが高品質で再生され、見かけ上の品質を向上させることができることが分かった。ただし、再生が一定のフレームレートで行われなため、落ち着かないという感想をもつ被験者もいた。重要なシーンを優先させる程度、バランスについてチューニングを行う必要があることがわかった。選択的マルチメディア通信方式は優先度付与の作業をユーザに負担させるという欠点がある。開発した優先度作成支援ツールはビデオ編集ツールの要領で、マウス操作によりシーンに優先度を与えることを可能にする。時間の短いコンテンツならば優先度付与の作業量、作業時間とも少なく、本方式が実用的であることが分かった。しかし、大量のコンテンツに優先度を与えるのはやはり無理があり、自動化に近いツールの支援が必要である。また、先読み方式については、シミュレーションにより転送が一時的に途切れたとしても、再

生を続行できることが分かった。しかし、先読み割り当ての適切な送信ビットレート、先読みする時間など、解析すべき項目が多く残っている。今後の課題としたい。

実時間指向優先転送方式に関しては、シミュレーションにより、低負荷時には実時間通信を達成できること、高負荷時には低優先度のパケットの実時間通信は達成できないかわりに、高優先度のパケットの実時間制約が守られることを確かめた。また、BSD/OS のカーネルに修正を加えて提案方式を組み込み、動作させた。実時間通信を実現するためのパケットのキューイング方式には複雑すぎて実装が困難なものが多いが、提案方式は大きな修正を必要とせず、実装可能であることが分かった。

提案した4つのマルチメディア通信方式の適用例としてモバイル電子会議システムを取り上げ、各方式をどのように利用できるか検討した。モバイル電子会議システムの一部、あるいはミドルウェアやネットワーク層のプロトコルとして、各通信方式を組み込むことで、モバイルコンピューティング環境におけるマルチメディア通信における問題の一部を解決できることが分かった。

さらに、実時間通信プロトコル設計言語を使って同期・非同期通信方式、実時間指向優先転送方式の記述を行い、シミュレータで動作確認、性能評価実験を行った結果、開発の早期に仕様の誤りを見つけたり、性能の予測をたてることができた。開発ツールの利用は、開発期間の短縮や、通信システムの効率の向上に役立つと思われる。

今後、IMT-2000 と呼ばれる次世代移動通信システムにおいて384Kbps から2Mbps という以前のLAN環境に近い高速無線データ通信サービスが広域で利用可能になり、現在の10倍以上の容量のマルチメディア情報を転送することが可能になる。また、通信コストの問題、サービスエリアの問題、バッテリーの問題も改善されると期待される。常時、端末がインターネットに接続し、切断するのはまれになるかもしれない。

しかし、ユーザの都合によりコミュニケーションの中断がおきるという問題は依然として残ると考えられ、同期・非同期通信方式の同期・非同期会議の会議形態の切り替えは有用であると思われる。また、無線リソースは依然として有限であり、効率的にリソースを利用する仕組みが必要である。選択的マルチメディア通信方式は、利用できる送信ビットレートをコンテンツの重要度に基づいて効率的に利用することができる。現在、インターネットやLANにおいて扱えるマルチメディア品質はテレビや映画には及ばず、単純に品質を比較するとその差は歴然である。一般のユーザに受け入れられるために、よりマルチメディア品質を向上させる必要がある。先読み方式や実時間指向優先転送方式の技術が品質向上に貢献できると考えている。

謝辞

本研究の過程において終始客観的な立場からの適切な御助言と親身な御鞭撻を賜り、研究をまとめるに当たり懇切なるご指導と励ましを頂いた、静岡大学情報学部情報科学科水野忠則教授に衷心より御礼申しあげる。

本論文をまとめる過程で種々な御指導と御鞭撻頂いた、静岡大学情報学部情報科学科中谷広正教授、富樫敦教授、渡辺尚助教授、静岡大学工学部電気・電子工学科橋本岳助教授並びに名古屋工業大学電気情報工学科田坂修二教授に対し、深謝申し上げます。また、研究の過程において、終始適切な御助言と励ましを頂いた佐藤文明助教授、太田剛講師、西垣正勝助手並びに愛知県立大学情報科学部井手口哲夫教授に感謝の意を表す。

本研究に対し御意見、御討論を頂いた三菱電機（株）名古屋製作所中野宣政博士、三菱電機（株）情報システム製作所宮西洋太郎博士、三菱電機（株）鎌倉製作所和田雄次博士、（株）インテックテクニカルサービスセンター田窪昭夫博士、三菱電機（株）情報技術総合研究所中村俊一郎博士、同所坂下善彦氏、三菱電機（株）情報システム製作所飯田登氏、三菱電機（株）情報技術総合研究所福岡久雄氏、通信・放送機構新川崎リサーチセンター辻順一郎氏、三菱電機（株）情報システム製作所青野正宏氏、三菱電機（株）情報技術総合研究所黒田正博氏、三菱電機（株）情報システム製作所横山繁盛氏並びに三菱電機（株）技術研究所吉田幸二氏に深謝申し上げます。

研究の過程では多くの同僚及び関係者の協力を得た。高品智一氏（現在、NTTコミュニケーションウェア（株））をはじめ、安藤努氏（同、（株）日立製作所）、河合東氏（同、三菱電機（株）鎌倉製作所）の水野研究室卒業生の諸氏、峰野博史氏、山田善大氏、増田彰久氏ら水野研究室の学生諸氏、並びに渡辺研究室、佐藤研究室、富樫研究室の学生諸氏に謝意を表す。

参考文献

- [1] 中島昭久, 貝山明. 移動通信ネットワークネットワークのエボリューション. 電子情報通信学会誌, Vol. 81, No. 4, pp. 380-388, April 1998.
- [2] 村田嘉利. 人を場所・時間的制約から解放するモバイルコンピューティング. 電子情報通信学会誌, Vol. 80, No. 8, pp. 844-849, August 1997.
- [3] 原田秀昭. モーバイルコンピューティング: 手のひらサイズの端末入門. NEC クリエイティブ, 1997.
- [4] 水野忠則, 太田賢. モーバイルコンピューティングの現状と将来. 電子情報通信学会誌, Vol. 80, No. 4, pp. 318-323, April 1997.
- [5] 水野忠則, 田窪昭夫. モーバイルコンピューティング. 情報処理学会誌, Vol. 36, No. 9, pp. 822-826, September 1995.
- [6] 斉藤忠夫, 立川敬二 (編). 移動通信ハンドブック. オーム社, 1995.
- [7] M. Dasen, G. Fankhauser, and B. Plattner. An Error Tolerant, Scalable Video Stream Encoding and Compression for Mobile Computing. *ACTS Mobile Summit 96*, pp. 762-771, November 1996.
- [8] A. Alwan, R. Bagrodia, N. Bambos, M. Gerla, L. Kleinrock, J. Short, and J. Villaseñor. Adaptive mobile multimedia networks. *IEEE Personal Communications*, Vol. 3, No. 2, pp. 34-51, April 1996.
- [9] 加藤正美, 臼井憲義, 田坂修二. PHSにおけるビデオ伝送方式の性能評価. 電子情報通信学会論文誌 B-II, Vol. J79-B-II, No. 10, pp. 646-656, October 1996.
- [10] ITU-T Recommendation H.263 Video coding for low bit rate communication, March 1996. http://www.itu.int/itudoc/itu-t/rec/h/h263__34484.html.
- [11] 山本吉伸, 仁木和久. 画像多重伝送方式によるインターネット動画サーバとビューア. 情報学ヒューマンインタフェース研報, Vol. 97, No. 2, pp. 33-38, January 1997.
- [12] J.M.F. Moura, R. S. Jasinschi, H. Shiojiri, and J. Lin. Video over wireless. *IEEE Personal Communications*, Vol. 3, No. 1, pp. 44-54, feb 1996.

- [13] 藤野信次, 徳世雅永, 飯田一郎. 無線エージェントと WWW プロキシ連携による適応的画質制御. 情処学モバイルコンピューティング研報, Vol. 98, No. 53, pp. 1-6, May 1998.
- [14] B. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol(RSVP)-Version 1 Functional Specification, September 1997. RFC2205.
- [15] S. Floyd and V. Jacobson. Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, Vol. 3, No. 4, pp. 365-386, August 1995.
- [16] Craig Partridge 著, 西田竹志訳. ギガビットネットワーク. SOFT BANK, 1995.
- [17] D.D. Clark, S. Shenker, and L. Zhang. Applications in an integrated services packet network: architecture and mechanism. In *Proc. ACM SIGCOMM'92*, pp. 14-26. ACM, 1992.
- [18] H. Kanakia, P. P. Misra, and A. R. Reibman. An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport. *IEEE/ACM Transactions on Networking*, Vol. 3, No. 6, pp. 671-682, December 1995.
- [19] Shanwei Cen, Calton Pu, Richard Staehli, Crispin Cowan, and Jonathan Walpole. A Distributed Real-Time MPEG Video Audio Player. *Fifth International Workshop on Network and Operating System Support of Digital Audio and Video (NOSSDAV'95)*, April 1995.
- [20] 橋本浩二, 知念正, 佐藤純, 柴田義孝. 圧縮ビデオデータ転送のためのパケット及びフレームレート制御法. 情報処理学会論文誌, Vol. 39, No. 2, pp. 337-346, February 1998.
- [21] T. Turetti and C. Huitema. Video conferencing on the internet. *IEEE/ACM Trans. Networking*, Vol. 4, No. 3, pp. 340-351, June 1996.
- [22] S. McCanne and V. Jacobson. vic: A flexible framework for packet video. *ACM Multimedia*, pp. pp. 511-522, November 1995.
- [23] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven Layered Multicast. *ACM SIGCOMM*, pp. 117-130, August 1996.
- [24] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: a transport protocol for real-time applications, November 1995. RFC1889.
- [25] 有馬秀平. 移動通信技術とネットワーク. 電子情報通信学会誌, Vol. 81, No. 4, pp. 355-361, April 1998.

- [26] 松下温, 重野寛. 無線 LAN の必要性和展望. 情報処理学会誌, Vol. 35, No. 12, pp. 1070-1079, December.
- [27] 大澤智喜. 高速無線 LAN 技術. 電子情報通信学会誌, Vol. 80, No. 12, pp. 1251-1254, December 1997.
- [28] C. Perkins. IP Mobility Supprt, October 1996. RFC2002.
- [29] 寺岡文男. ホスト移動透過性を提供するプロトコル. 情報処理学会誌, Vol. 35, No. 12, pp. 1093-1098, December 1994.
- [30] R. Droms. Dynamic Host Configuration Protocol, October 1993. RFC1541.
- [31] 重野寛, 清松和明, 岡田謙一, 松下温. モバイル・コンピューティングをサポートする MC^2 Platform. 電子情報通信学会論文誌, No. 2, pp. 49-57, February 1998.
- [32] 徳升厚美, 広川賢, 高橋成文. 移動体通信ミドルウェア「モバイルコンポーネント」. 情処学モバイルコンピューティング研究グループ研報, Vol. 96, pp. 39-43, July 1996.
- [33] 峰野博史, 池谷利明, 山田善大, 太田賢, 水野忠則. モバイル電子会議システム PAR-CAE の提案. 情報処理学会全国大会講演論文集, pp. 3-543-544, May 1997.
- [34] James J. Kistler and M. Satyanarayanan. Disconnected Operation in the CODA File System. In T. Imielin'ski and H.F. Korth, editors, *Mobile Computing*, pp. 537-570. Kluwer Academic Publishers, Boston, 1996.
- [35] M. Satyanarayanan, James J. Kistler, Lily B. Mummert, Maria R. Ebling, Puneet Kumar, and Qu Lu. Experience with disconnected operation in a mobile computing environment. In T. Imielin'ski and H.F. Korth, editors, *Mobile Computing*, pp. 537-570. Kluwer Academic Publishers, Boston, 1996.
- [36] 宮西洋太郎, 中村健二, 佐藤文明, 水野忠則. ノマディック環境における複製データの管理. 情処学モバイルコンピューティング研究グループ研報, Vol. 96, pp. 15-20, July 1996.
- [37] J. Postel. Internet Protocol-DARPA Internet Program Protocol Specification, September 1981. RFC791.
- [38] 辻順一郎, 小津浩直, 三浦敦史, 滝沢直樹, 水野忠則. モバイルプロキシサーバシステムの試作. 情報処理学会研究報告, Vol. 97, No. 72, pp. 67-72, July 1997.
- [39] 鶴保証城. NTT データにおけるモバイルコンピューティングの取り組み. 情処学モバイルコンピューティング研究グループ研報, Vol. 96, No. MBL-1, pp. 1-2, July 1996.

- [40] 植原啓介, 西村厚, 村井純. LWPA: インターネット環境における広域無線通信メディア利用のためのアーキテクチャ. インターネットカンファレンス'97 論文集, pp. 49-62, December 1997.
- [41] M. Kojo, K. Raatikainen, and T. Alanko. Connecting mobile workstations to the internet over a digital cellular telephone network. In T. Imielin'ski and H.F. Korth, editors, *Mobile Computing*, pp. 537-570. Kluwer Academic Publishers, Boston, 1996.
- [42] 田中利憲. PHS によるデータ通信. 情報処理学会誌, Vol. 38, No. 5, pp. 428-434, May 1997.
- [43] K.R. Rao, J.J.Hwang 著, 安田浩, 藤原 洋監訳. デジタル放送・インターネットのための情報圧縮技術. 共立出版 bit 別冊, 東京都, 1998.
- [44] 村田正幸. マルチメディアコンピュータネットワークの通信品質保証. 電子情報通信学会誌, Vol. 81, No. 4, pp. 362-370, April 1998.
- [45] 柴田義孝, 橋本浩二, 渡辺光輝. 分散マルチメディアシステムにおける QoS 機能. 情報処理学会論文誌, Vol. 37, No. 5, pp. 731-739, May 1996.
- [46] Mark E. Perkins, Keith Evans, Dominique Pascal, and Leigh A. Thorpe. Characterizing the Subjective Performance of the ITU-T 8 kb/s Speech Coding Algorithm — ITU-T G.729. *IEEE Communications Magazine*, pp. 74-81, September 1997.
- [47] Richard V. Cox and Peter Kroon. Low bit-rate speech coders for multimedia communication. *IEEE Communications Magazine*, pp. 34-41, December 1996.
- [48] S. Deering. Internet Protocol, Version 6 (IPv6) Specification, December 1995. RFC1883.
- [49] S. Deering. Host Extensions for IP Multicasting, August 1989. Internet RFC 1112.
- [50] 渡部和雄, 坂田史郎, 前野和俊, 福岡秀幸, 大森豊子. マルチメディア分散在籍会議システム MERMAID. 情報処理学会論文誌, Vol. 32, No. 9, pp. 1200-1209, September 1991.
- [51] Ronald J. Vetter. Videoconferencing on the internet. *computer*, Vol. 28, No. 1, pp. 77-80, January 1995.
- [52] 太田賢, 渡辺尚, 水野忠則. モバイルユーザのための同期 / 非同期電子会議システム. 情処学モバイルコンピューティング研究グループ研報, Vol. 96, pp. 13-18, November 1996.
- [53] 星徹, 松井進, 高田治, 岩見直子, 高原桂子, 小山俊明. リアルタイム・蓄積統合マルチメディアコミュニケーションシステムト・プラットフォーム. 情処学グループウェア研報, Vol. 95, No. 87, pp. 25-30, October 1995.

- [54] 太田賢, 山田善大, 水野忠則. モーバイルコンピューティング環境における協調作業を支援する電子会議システム. *情報学モーバイルコンピューティング研報*, Vol. 98, No. 53, pp. 41-48, May 1998.
- [55] 戸田浩. サウンドエフェクト. *C MAGAZINE*, Vol. 8, No. 12, pp. 22-50, December 1996.
- [56] T. Imielinski and H.F. Korth. *Mobile Computing*. Kluwer Academic Publishers, Boston, 1996.
- [57] 呉世雄, 藤川和利, 松浦敏雄, 下條真司, 宮原秀夫. 分散型マルチメディアシステム Symphony における QoS 指定を含んだシナリオ記述言語. *電子情報通信学会論文誌 B-I*, Vol. J79-B-I, No. 5, pp. 329-337, May 1996.
- [58] Anup Rao and Rob Lanphier. Real Time Streaming Protocol (RTSP), November 1996.
<http://www6.realaudio.com/prognet/rt/protocol.txt>.
- [59] M. Reisslein and K. W. Ross. A Join-the-Shortest-Queue Prefetching Protocol for VBR Video on Demand. In *Proceedings of IEEE International Conference on Network Protocols*.
- [60] K. M. Curewitz, P. Krishnan, and J. S. Vitter. Practical Prefetching via Data Compression. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD '93)*, pp. 257-266, May 1993.
- [61] Nobumasa Nakano. Time critical communication architecture in factory automation. In *JSPE-IFIP WG5.3 Workshop on THE DESIGN OF INFORMATION INFRASTRUCTURE SYSTEMS FOR MANUFACTURING*, pp. 241-252, 1993.
- [62] C. Topolcic, S.Casner, C.Lynn, P.Park, and K.Schroder. Experimental internet stream protocol, version 2(ST-II), October 1990. RFC1190.
- [63] W. T. Strayer, B. J. Demsey, and A. C. Weaver. *The Xpress Transfer Protocol*. Addison Wesley.
- [64] K. Ohta, T. Watanabe, and T. Mizuno. A proposal of network protocol with performance for multimedia communication system. *IEICE TRANS.INF. & SYST.*, Vol. E79-D, No. 6, pp. 719-727, July 1996.
- [65] K. Ohta, Y. Sakashita, T. Watanabe, and T. Mizuno. A proposal of network protocol with performance for multimedia communication system. In *Second Joint Workshop on Multimedia Communications*.

- [66] K. Ohta, Y. Sakashita, T. Watanabe, and T. Mizuno. Time critical network protocol. In *The 10th International Conference on Information Networking(ICOIN-10)*.
- [67] 日経コミュニケーション 1998.4.6. インターネット技術最前線. April 1998.
- [68] David L. Mills. Precision synchronization of computer network clocks. *Computer Communication Review*, Vol. 24, No. 2, pp. 28-43, April 1994.
- [69] 河合東, 太田賢, 渡辺尚, 水野忠則. ネットワーク応答特性プロトコルの実装と評価. 情報処理学会研究報告, Vol. 96, No. 40, pp. 145-150, May 1996.
- [70] 河合東, 太田賢, 渡辺尚, 水野忠則. 広域網におけるネットワーク応答特性プロトコルの実装と評価. 情報処理学会研究報告, Vol. 97, No. 57, pp. 31-36, June 1997.
- [71] Sudhir R. Ahuja and J. Robert Ensor. Coordination and control of multimedia conferencing. *IEEE Communications Magazine*.
- [72] William J. Clark. Multipoint multimedia conferencing. *IEEE Communications Magazine*.
- [73] 菅沼拓夫, 藤田茂, 菅原研次, 木下哲夫, 白鳥則郎. マルチエージェントに基づくやわらかいビデオ会議システムの設計と実装. 情報処理学会論文誌, Vol. 38, No. 6, pp. 1214-1224, June 1997.
- [74] George Xylomenos and George C. Plyzos. IP Multicast for Mobile Hosts. *IEEE Communications Magazine*.
- [75] R. Kazman, R. Al-Halimi, W. Hunt, and M. Mantei. Four paradigms for indexing video conferences. *IEEE MULTIMEDIA*, Vol. 3, No. 1, pp. 63-73, 1996.
- [76] 水野忠則. 通信システムの形式記述法の標準化. 情報処理学会誌, Vol. 31, No. 1, pp. 2-81, January 1990.
- [77] P. H. J. van Eijk, C. A. Vissers, and M. Diaz. *THE FORMAL DESCRIPTION TECHNIQUE LOTOS*. ELSEVIER SCIENCE PUBLISHERS B.V., 1989.
- [78] D. Fernandez, E. Vazquez, and J. Vinyes. IO: An Estelle simulator for performance evaluation. In *FORTE*, pp. 53-65.
- [79] Roberto Saracco, J. R. W. Smith, and Rick Reed. *Telecommunications Systems Engineering using SDL*. ELSEVIER SCIENCE PUBLISHERS B.V., 1989.
- [80] 辻宏郷, 佐藤文明, 勝山光太郎, 水野忠則, 曾我正和. 形式手法による通信ソフトウェア試験データ生成とその試験法. 情報処理学会論文誌, Vol. 34, No. 6, pp. 1347-1360, June 1993.

- [81] 厚井裕司, 妹尾尚一郎, 高田佳典, 井手口哲夫, 石坂充弘. LANにおけるタイムクリティカル通信プロトコルの提案と評価. 電子情報通信学会論文誌, No. 8, pp. 882-891, November 1993.
- [82] Initial draft on enhancements to LOTOS, nov 1993. Part 6 ISO/IEC JTC 1/SC 21 N8023.
- [83] Ashley Mc Clenaghan. Mapping time-extended LOTOS to standard LOTOS. In *FORTE*, pp. 239-254.
- [84] Guy Leduc. An upward compatible timed extension to LOTOS. In *FORTE*, pp. 223-238.
- [85] 中田明夫, 東野輝夫, 谷口健一. 隣接しない動作間の時間制約を記述するための LOTOS 言語の拡張とその等価性の検証. 情報処理学会マルチメディア通信と分散処理研究会, Vol. 61, No. 19, pp. 139-146, July 1993.
- [86] Gerard J. Holzmann. *Design and Validation of Computer Protocols*. PRENTICE HALL, 1991.
- [87] Ken Ohta, Nobumasa Nakano, Sanshiro Sakai, Takashi Watanabe, and Tadanori Mizuno. Time oriented protocol testing simulator. In *Proceedings of 7th IFIP WG6.1 International Workshop on Protocol Test Systems(IWPTS VII)*, pp. 307-314, 1 1994.
- [88] 太田賢, 中野宣政, 渡辺尚, 水野忠則. プロトコル記述言語 PROMELA の時間特性の拡張. 情報処理学会全国大会講演論文集, pp. 1-221-222, May 1994.
- [89] 太田賢, 渡辺尚, 水野忠則. 時間変化要素を含む分散型通信システムの記述法. 情報処理学会研究報告, Vol. 95, No. 13, pp. 13-18, January 1995.

筆者発表論文

A 学術雑誌等に発表した論文

1. 中野 宣政, 太田 賢, 渡辺 尚, 水野 忠則 : 分散 TCCS 実現のためのグループ管理機能、情報処理学会論文誌, Vol.36, No.8, pp.2027-2036(1995.8).
2. K.Ohta, T.Watanabe, and T.Mizuno : A Proposal of Network Protocol with Performance for Multimedia Communication System, IEICE transaction on Informaton and Systems, Vol.E79-D, No.6, pp.719-727(1996.6).
3. 坂下 善彦, 太田 賢, 水野 忠則 : 制御システムの自立協調機構, 静岡大学大学院電子科学研究科, pp.135-140(1997.3).
4. 水野 忠則, 太田 賢 : モバイルコンピューティングの現状と将来, 電子情報通信学会誌, Vol.80, No.4, pp.318-323(1997.5).
5. 太田 賢, 渡辺 尚, 水野 忠則 : ワイヤレス通信環境における選択的マルチメディア通信方式の実装, 情報処理学会論文誌, Vol.39, No.2, pp.312-320(1998.2).
6. 太田 賢, 山田 善大, 奈良岡 将英, 渡辺 尚, 水野 忠則 : モバイルコンピューティング環境における協調作業を支援する電子会議システム, 情報処理学会論文誌, Vol.39, No.10, pp.2879-2887(1998.10).

B 国際会議に発表した論文

1. K.Ohta, N.Nakano, S.Sakai, T.Watanabe, and T.Mizuno : Time Oriented Protocol Testing Simulator, Proceedings of 7th IFIP WG6.1 International Workshop on Protocol Test Systems(IWPTS VII), pp.307-314(1994.11).
2. N.Nakano, K.Ohta, S.Sakai, T.Watanabe, and T.Mizuno : Development of Group Management Functions and Mechanisms for Distributed TCCS and its Validating Environment applying LOTOS-T, Proceedings of IEEE International Conference on Information Network(ICOIN-9), pp.93-98(1994.12).
3. K.Ohta, Y.Sakashita, T.Watanabe, and T.Mizuno : A Proposal of Network Protocol with Performance for Multimedia Communication System, Second Joint Workshop on Multimedia Communications, pp.4-1-1-4-1-8(1995.10).
4. K.Ohta, Y.Sakashita, T.Watanabe, and T.Mizuno : Time Critical Oriented Network Protocol, Proceedings of the 10th International Conference on Information Networking ICOIN-10, pp.268-274(1996.1).
5. K.Ohta, T.Watanabe, and T.Mizuno: Selective Multimedia Access Protocol for Wireless Multimedia Communication, 1997 Pacific Rim Conference on Communications, Computers and Signal Processing(PACRIM'97) Conference Proceedings Volume 1 of 2, pp.81-84(1997.8).

6. Y.Sakashita, K.Ohta, F.Sato and T.Mizuno : On the Notion of Propagation of Control in Collaborative and Autonomous Mechanism, Proc.of ISCOM97, pp.158-161(1997.12).
7. Y.Sakashita, K.Ohta, F.Sato and T.Mizuno : On the Notion of Adaptability to the environments in Autonomous Mechanism, The 12th International Conference on Information Networking, pp.705-708(1998.1).
8. Y.Yamada, K.Ohta, and T.Mizuno: Extracting and viewing information method for mobile teleconference system, Asia Pacific Computer Human Interaction(APCHI'98), pp.430-435(1998.7).
9. H.Mineno, K.Ohta, M. Aono, and T.Mizuno:Proposal of Protocol for Sharing Multiple Paths in Cluster Type Network, to be appeared in Proceedings of 5th International Workshop on Mobile Multimedia Communications MoMuC '98(1998.10).
10. A.Masuda, K.Ohta, and T.Mizuno: Implement of Network Service for Selective Multimedia Access Protocol, to be appeared in Proceedings of The 4th Asia-Pacific Conference on Communications and 6th Singapore International Conference on Communication Systems(1998.11).

C 全国大会, 研究会において発表した論文

1. 太田 賢, 中野 政, 渡辺 尚, 水野 忠則:‘プロトコル記述言語PROMELAの時間特性の拡張’, 情報処理学会全国大会講演論文集, pp.1-221-222(1994.3).
2. 中野 宣政, 太田 賢, 渡辺 尚, 水野 忠則 : TCCA仕様記述対応LOTOSの時間拡張と、その実行環境について, 情報処理学会研究報告, 情処研報 Vol.94, No.56, pp.43-48(1994.7).
3. 中野 宣政, 安藤 勉, 太田 賢, 渡辺 尚, 水野 忠則 : TCCS仕様記述対応LOTOS-T実行環境構築をベースとするソフトTCCSメカニズムの研究, マルチメディア通信と分散処理ワークショップ, 情処ワークショップ論文集 Vol.94, No.1, pp.11-20(1994.10).
4. 太田 賢, 渡辺 尚, 水野 忠則 : ‘時間変化要素を含む分散型通信システムの記述法’, 情報処理学会研究報告, 情処研報 Vol.95, No.13, pp.13-18(1995.1).
5. 中野 宣政, 安藤 勉, 太田 賢, 佐藤 文明, 水野 忠則 : ソフトTCCSとその開発検証システムの構築について, 情報処理学会研究報告, 情処研報 Vo.95, No.72, pp.37-42(1995.3).
6. 坂下 善彦, 太田 賢, 水野 忠則 : 予約アクセスによる時間同期処理の試み, 情報処理学会研究報告, 情処研報 Vol.95, No.53, pp.129-134(1995.5).

7. 中野 宣政, 安藤 勉, 太田 賢, 佐藤 文明, 水野 忠則: ソフトTCCSオブジェクトクラスSTOCについて, 情報処理学会研究報告, 情処研報 Vol.95, No.53, pp.69-74(1995.5).
8. 太田 賢, 坂下善彦, 水野忠則: 'ネットワーク応答特性プロトコルの提案とマルチメディア通信システムへの適用', 情報処理学会研究報告, 情処研報 Vol.95, No.61, pp.43-48(1995.7).
9. 河合 東, 太田 賢, 渡辺 尚, 水野 忠則: ネットワーク応答特性プロトコルの実装と評価, 情報処理学会研究報告, 情処研報 Vol.96, No.40, pp.145-150(1996.5).
10. 太田 賢, 渡辺 尚, 水野 忠則: ワイヤレス通信環境における蓄積型マルチメディア通信方式, 情処研報 Vol.96, No.MBL-1, pp.69-74(1996.7).
11. 坂下 善彦, 池田 峰次, 太田 賢, 水野 忠則: 共同作業と協調作業への考察, マルチメディア通信と分散処理ワークショップ, 情処ワークショップ論文集 Vol.96, No.2, pp.373-378(1996.10).
12. 太田 賢, 渡辺 尚, 水野 忠則: モバイルユーザのための同期 / 非同期電子会議システム, 情処研報, Vol.96, No.MBL-3, pp.13-18(1996.11).
13. 池谷 利明, 峰野 博史, 山田 善大, 太田 賢, 水野 忠則: モバイル電子会議システム PARCAE におけるクライアントの実現, 情報処理学会全国大会講演論文集, pp.3-533-3-534(1997.3).
14. 山田 善大, 池谷 利明, 峰野 博史, 太田 賢, 水野 忠則: モバイル電子会議システム PARCAE におけるサーバの実現, 情報処理学会全国大会講演論文集, pp.3-541-3-542(1997.3).
15. 峰野 博史, 池谷 利明, 山田 善大, 太田 賢, 水野 忠則: モバイル電子会議システム PARCAE の提案, 情報処理学会全国大会講演論文集, pp.3-543-3-544(1997.3).
16. 太田 賢, 渡辺 尚, 水野 忠則: ワイヤレス通信環境における選択的マルチメディア通信方式の実装, 情処研報 97-DPS-82, Vol.97, No.35, pp.141-146(1997.4).
17. 河合 東, 太田 賢, 渡辺 尚, 水野 忠則: 広域網におけるネットワーク応答特性プロトコルの実装と評価, 情報処理学会研究報告, Vol.97, No.57, pp.31-36(1997.6).
18. 太田 賢, 渡辺 尚, 水野 忠則: モバイル会議におけるユーザ支援システムの設計, 情処ワークショップ論文集, Vol.97, No.2, pp.215-220(1997.7).
19. 山田 善大, 太田 賢, 水野 忠則: 蓄積型モバイル電子会議システムにおける情報抽出・参照方式, 情報処理学会研究報告, Vol.97, No.72, pp.73-78(1997.7).
20. 太田 賢, 渡辺 尚, 水野 忠則: コンテンツ指向モバイルマルチメディアアクセス方式の実現, 情報処理学会第 55 回 (平成 9 年後期) 全国大会講演論文集 (3), pp.3-618-619(1997.9)
21. 峰野 博史, 太田 賢, 青野 正宏, 水野 忠則: クラスタ型ネットワークにおける通信回線共有プロトコルの提案, 情報処理学会第 55 回全国大会講演論文集 (3), pp.3-616 - 3-617(1997.9).

22. 青野 正宏, 峰野 博史, 太田 賢, 渡辺 尚, 水野 忠則: クラスタ型モバイル通信の効果, 情報処理学会研究報告, pp.57-62(1997.12).
23. 太田 賢, 増田 彰久, 渡辺 尚, 水野 忠則: コンテンツ指向モバイルマルチメディアアクセス方式の解像度の制御, 情処研報 97-MBL-3, Vol.97, No.117, pp.69-74(1997.12).
24. 増田 彰久, 太田 賢, 渡辺 尚, 水野 忠則: 選択的マルチメディア通信方式 SMAP ネットワーク機能の実装, 全国大会講演論文集 (3), pp.506-507(1998.3).
25. 奈良岡 将英, 山田 善大, 太田 賢, 渡辺 尚, 水野 忠則: モバイル電子会議システム Dynamite における音声処理機能の実装, 全国大会講演論文集 (3), pp.358-359(1998.3).
26. 太田 賢, 山田 善大, 水野 忠則: モバイルコンピューティング環境における協調作業を支援する電子会議システム, 情処研報 98-MBL-5, Vo.98, No.53, pp.41-48(1998.3).
27. 増田 彰久, 太田 賢, 水野 忠則: 選択的マルチメディア通信方式 SMAP のネットワークサービスの実装, 情処研報 98-MBL-5, Vol.98, No.53, pp.55-62(1998.5).
28. 峰野博史, 太田 賢, 青野正宏, 水野忠則: クラスタ型ネットワークにおける通信回線共有方式の実装, 情処シンポジウムシリーズ, Vol.98, No.8, pp.677-684(1998.7).
29. 太田 賢, 増田彰久, 渡辺 尚, 水野 忠則: ワイヤレスマルチメディア通信における QoS を保証する先読み方式, 情処シンポジウムシリーズ, Vol.98, No.8, pp.685-692(1998.7).

D 表彰

1. 第 55 回情報処理学会全国大会大会奨励賞: コンテンツ指向モバイルマルチメディアアクセス方式の実現, 情報処理学会第 55 回 (平成 9 年後期) 全国大会講演論文集 (3), pp.3-618-619(1997.9)
2. 電子情報通信学会東海支部学生研究奨励賞: モバイルコンピューティング環境におけるマルチメディア通信方式に関する研究 (1998.5).