

QOS に基づいたマルチメディア OS

岡村耕二 稲垣英太郎 吉川耕平
松尾聡 田中裕之 荒木啓二郎
奈良先端科学技術大学院大学 情報科学研究科

我々は、マルチメディアコミュニケーションツールを用いてコミュニケーションを行なう場合、システムは、「単一メディアの品質」と「メディア間同期の品質」を保証する必要があると考えた。我々が現在、設計、開発中のマルチメディアオペレーティングシステム PDE-II は、これらの品質の保証をしてマルチメディア処理を行なうことができる。

本稿では、まず、これらのサービスの品質の定義を行ない、その品質に基づいたマルチメディア処理モデルの提案を行なう。次に、そのモデルに基づいたマルチメディアオペレーティングシステム PDE-II の概要を述べる。

Multimedia Operating System based on QOS

Koji OKAMURA, Eitaro INAGAKI, Kouhei YOSHIKAWA,
Satoshi MATSUO, Hiroyuki TANAKA and Keijiro ARAKI
Graduate School of Information Science
Nara Institute of Science and Technology

We notice that quality of a single medium and synchronization between multimedia are very important for communication on multimedia communication tools. We are designing and developing Multimedia Operating System PDE-II. PDE-II will be able to guarantee these qualities for multimedia processing.

In this paper, firstly we define these qualities and introduce a model for multimedia processing based on the qualities. Then we describe about the Multimedia Operating System PDE-II based on the model.

1 はじめに

近年の計算機の高速化および、ネットワークの大容量化によって、ネットワークで接続された計算機間でヒューマンコミュニケーションを行なうことができるマルチメディアコミュニケーションツールの利用が実用的になりつつある [1]。

マルチメディアを扱うマルチメディアコミュニケーションツールを支援するために、システムはメディアの持つ連続性によって各メディアを連続的に処理する必要があり、また、マルチメディアの持つ特性によって互いに関係付けられた複数メディアを同期させながら処理する必要がある。

このようなマルチメディア処理を行なうために必要な機能は従来のシステムには備わっていない。そこで、我々は、基本ソフトウェアであるオペレーティングシステムレベルでマルチメディアに必要な機能を提供させるために、オペレーティングシステムのスケジューリング方式、資源管理方式および入出力管理方式に時間的制約や同期の保証を行なう機能を組み込む研究を行なっている。

本稿では、2章でマルチメディア処理におけるメディアの連続性とメディア間に関係付けを明確に定義し、それらの実現を行なうためにシステムが保証すべきサービスの品質の定義を行なう。3章では、そのサービスの品質に基づいたマルチメディア処理モデルの提案を行ない、4章で、我々が現在開発中であるマルチメディアオペレーティングシステム PDE-II の概要を述べ、最後に本稿のまとめを行なう。

2 マルチメディア処理

メディアとは、情報が知覚され、表現され、蓄積され、または、伝えられる手段であり、その性質や機能によっていくつかの種類に分類することができる [7]。マルチメディア処理には、複数種類のこれらのメディアを扱う特性が含まれる。本章では、まず、これらの各メディアの特性を説明し、次にメディアの連続性とメディアを関連つけたマルチメディアの処理について説明する。

2.1 メディア処理

メディアは、例えば、音声や映像といった情報そのものであるメディアと、情報そのものであるメディアのために使われる例えばスピーカやモニタといった物理的手段であるメディアに大別することができる。以下に各メディアの説明を行なう。

・ 情報そのものであるメディア

情報そのものであるメディアのうち、ユーザによって知覚されるメディアは知覚メディアと呼ばれ、計算機間で相互交換されたり、蓄積されたりするために符号化された形式で記述されるメディアは表現メディアと呼ばれる。

知覚メディア (perception media)

知覚メディアは音声や映像といったユーザによって知覚されるメディアである。

表現メディア (representation media)

表現メディアは ASCII といった相互交換されるデータの種類や、MPEG 標準といった符号化された形式で記述されるメディアである。

・ 物理的手段であるメディア

物理的なメディアのうち、ユーザに対して知覚メディアの入出力を行なう、例えばスピーカやモニタのようなメディアは表示メディアと呼ばれ、計算機の内部で表現メディアを扱う、例えばケーブルやディスクのようなメディアは相互交換メディアと呼ばれる。

表示メディア (presentation media)

表示メディアはスピーカやモニタのようにユーザに対して情報を再現させたり（出力装置）、マイクやカメラのようにユーザからの情報を取り込んだり（入力装置）するメディアである。

相互交換メディア (interchange media)

相互交換メディアは、データを相互交換するためのメディアであり、伝送メディアと蓄積メディアがある。

伝送メディア (transmission media)

伝送メディアは例えばケーブルのように表現メディアを伝送するためのメディアである。

蓄積メディア (storage media)

蓄積メディアは例えばディスクのように表現メディアを蓄積するためのメディアである。

図 2-1 に、各メディアの関係を示す。

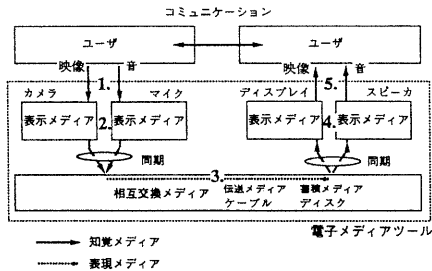


図 2-1: メディア処理における各メディアの関係

図 2-1 中の矢印はメディア処理の流れを表している。各処理を以下に説明する。

1. 表示メディア上で知覚メディアがとらえられる。
2. 知覚メディアが表現メディアに符号化される。
3. 表現メディアが、相互交換メディアで、伝送、蓄積される。
4. 伝送されてきた表現メディアが知覚メディアに復号化される。
5. 知覚メディアが表示メディア上で再生される。

2.2 マルチメディア処理

知覚メディアが表現メディアに変換される時、知覚メディアの持つ連続性の性質によって、表現メディアは時系列で連続的に定義されている。また、マルチメディアとは、複数種類の表現メディアを扱う特性であることから、関連する表現メディアは、互いに関係付けられている。

ところで、マルチメディアコミュニケーションツールでマルチメディア処理が行なわれる時、各メディアの連続性やメディア間の関連付けが損なわれると、コミュニケーションにおける意味を正確に伝えることができない [13]。例えば、音声と映像を用いて意図を伝える場合は表示メディアにとらえられた時と同じタイミングで再生を行なわなければ、その意図は正確に伝わらない。そのため、システムは、これらのマルチメディア処理を行なう場合、サービスの品質 (QOS=Quality of Service) の保証を行なう必要がある。

マルチメディア処理のサポートをするシステムは表示メディアや相互交換メディアといった電子メディアであるハードウェアとそれらの制御を行なうソフトウェアで構成される。その電子メディアを用いたアプリケーションの品質は、表示メディアの解像度、知覚メディアと表現メディア間の変換速度、伝送メディアの速度、

容量および、蓄積メディアへのアクセス速度といったそれぞれの電子メディアが提供する QOS に依存する。

また、これらの電子メディアが複数のユーザによって使用される場合は、資源の管理を行なうオペレーティングシステムがそれらの QOS の調停を行なう必要がある。

3 マルチメディア処理モデル

我々は、マルチメディア処理を行なう時、特にサービスの品質として、「単一メディアの品質」と「メディア間同期の品質」を保証する必要があることに着目した [8]。単一メディアの品質とは個々のメディア情報がいかに正確に伝達されるかということに関連し、メディア間同期は、複数のメディアが、送信側といかに同じようなタイミングで再生されるかということに関連する。

3.1 サービスの品質

我々が定義した、単一メディアの品質とメディア間同期の品質について説明を行なう。

単一メディアの品質

単一メディアの品質は、単位時間あたりに処理されるメディアの単位数である時間的解像度と、そのメディアの一単位の品質である空間的解像度で表現することができる [2]。

例えば、音声の時間的解像度はサンプリング周波数であり、空間的解像度は量子化ビット数である。一方、映像の時間的解像度は単位時間あたりに処理される画像のフレーム数であり、空間的解像度はフレームのサイズやピクセル数である。

メディア間同期の品質

互いに関連のある複数の知覚メディアが、表示メディアでとらえられた時と同じタイミングで別の表示メディアで再生されるためにはメディア間の同期を取る必要がある。

しかしながら、受信側で送信時と全く同じタイミングで、複数メディアの再生を行なうことは困難であるため、同期には「ずれ」が生じる。アプリケーションによって許容できる同期のずれの度合が異なる。我々は、このようなメディアの同期のずれの度合を品質として定義した。

3.2 マルチメディア処理モデル

図 3-1 に我々が提案する単一メディアの品質とメディア間同期の保証を行なうマルチメディア処理モデル

ルを示す [8]。

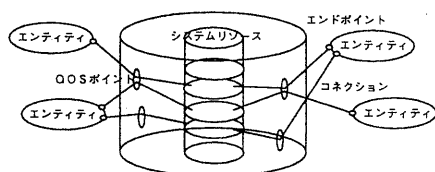


図 3-1: マルチメディア処理モデル

マルチメディア処理モデルは、エンティティ、システムリソース、エンドポイント、コネクションおよび QOS ポイントで構成される。各構成要素の説明を行なう。

エンティティ

エンティティとはプロセスやデバイスなどマルチメディアデータの通信および処理を行なう主体である。

データの入出力口であるエンドポイントを任意個もつことができる。

システムリソース

システムリソースは、エンティティ間で通信を行なうために必要なシステムのリソースを抽象化したもので、それを共通に使用するエンティティの集合に対して一つ存在するものである。

システムリソースは有限で、個々のコネクションはその設定に必要な品質から計算できる量を消費する。

エンドポイント

エンドポイントは、エンティティのデータの出入口であり、そこを流れる単一メディアの品質を設定できる。

コネクション

コネクションはエンティティ間でデータの交換を行なうための仮想伝送路であり、送信/受信両方のエンドポイントおよび QOS ポイントを生成する。生成の際、両ポイントに設定された単一メディアの品質を保証するために必要なリソース量をシステムリソース内に確保する。

QOS ポイント

QOS ポイントは、保証する同期の品質を設定して生成され、単一コネクション上でのデータ到着の時間的なずれの補正と、コネクション間の同期を行なう。

個々のコネクションについて、入力/出力の両方で独立に同期を設定できるものとし、また、各々の QOS ポイントでは、そこを通る複数のコネクションの優先度に従って同期の品質の保証を行なう。

4 PDE-II の概要

我々は、3 章で提案したマルチメディア処理モデルに基づいて、マルチメディアオペレーティングシステム PDE-II の設計、試作を行なっている。4 章では、PDE-II の中心的機能であるリソース管理機構、メディア同期機構、周期スレッドおよびエンティティ管理機構について説明を行ない、PDE-II 上のアプリケーションについても述べる。

4.1 リソース管理機構

メディアの連続性やメディア間同期の品質の保証を行なうために、オペレーティングシステムはリソースの管理を行なう必要がある [3]。リソース管理機構では、コネクションで接続されたエンティティの組に対してリソースの予約確保を行ない、マルチメディア処理の品質の保証を目指す。現在、我々は予約確保を行なうリソースとして、CPU とネットワークを想定している。[11]。以下に CPU とネットワークのそれぞれの予約確保の定義を示す。

CPU

あるプロセスに割り当てられる CPU の時間は、単位時間あたり一定の決められた時間である必要がある。

ネットワーク

あるコネクションは、単位時間あたり一定の決められた量のデータの送受信が行なえる必要がある。

我々は、まず、各リソースの定量化を行ない、図 4-1 に示すように、定量化されたリソースの管理を各リソースごとに存在用意しているリソースサーバに行なわせる。また、同一コネクションにおいて、CPU とネットワークの利用は互いに独立ではないため、リソースサーバは他のリソースサーバと調停を行ないながらリソースの管理を行なっている。

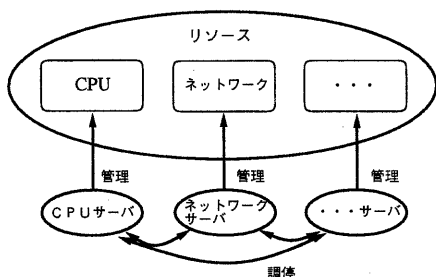


図 4-1: リソースサーバ

現在、CPU の基本的な予約確保を行なう実時間ラウンドロビン機構の実装を行なっている [11]。ネットワークの予約確保には、帯域確保のできる ST-II プロトコル [4] の導入を検討している。

ところで、コネクションに一定のリソースが予約確保されても、そのエンティティ間の通信のために常にリソースが 100% 使用されているとは限らない。例えば、音声通信では無音部の送信は行なわない場合が多いし、映像通信の場合は差分のみの送信を行なう場合が多いからである。また、リソースを予約確保することによってシステム内のリソースの使用率を落すことにもなりかねない。

そこで、我々は、システム内でリソースを仮想リソースとして扱う。仮想リソースを用いることによって、実際の物理的なリソースの総量よりも多くのリソースの予約確保を行なうことが可能になり、システム内のリソースを有効に利用することができる。

4.2 メディア同期機構

QOS ポイントは、マルチメディアアプリケーションから時間依存のデータ処理を行なう部分だけを分離し、これを OS の機能として提供するものである。

これにより、アプリケーションプログラムのうち時間に依存しない処理を行なう部分だけがエンティティとして独立したものとなり、プログラマは QOS ポイントとの間の API を用いてエンティティのプログラミングをすることで連続メディアを扱うアプリケーションを記述することができる。

コネクションが張られたエンティティ間では、時系列で定義された連続メディアの送受信が行なわれる。QOS ポイントでは、時間的解像度に基づいた一周期のデータ処理が完了すると一周期の時間だけ進む論理時間 [5] を用いてこのエンティティ間の時間的制約のあるデータの送受信の保証を行なう。

図 4-2 に論理時間を用いた同期処理モデルを示す。

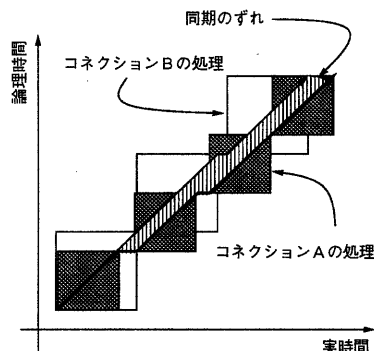


図 4-2: 論理時間を用いた同期処理モデル

図 4-2 のブロックは、あるコネクションで一周期時間当たりに行なう連続メディアデータの処理を表し、ブロックの一边は周期時間の長さを表す。ブロックの左端はエンティティがデータ処理を開始した時間であり、右端はデータ処理のデッドラインである。ある実時間において、あるコネクションにおける処理と別のコネクションでの処理の論理時間の差を同期のずれとする。われわれはこの同期のずれの許容範囲を「同期の品質」と定義した。

図 4-2 において、コネクション B は、コネクション A に同期しているとすると、この時、「コネクション B は、コネクション A に従う」と呼ぶ。この場合の同期の品質は、「コネクション A に対するコネクション B の同期の品質」とする。同期の品質は、データスキップと遅延挿入によって調整され保証される。

データスキップ

コネクション B の処理の進行がコネクション A の処理より遅れ過ぎて同期の品質を超過した場合、同期機構はコネクション B のデータを捨てる。このようにコネクション B の論理時間の進行を一時的に加速することで同期のずれを修正する処理をデータスキップと呼ぶ。

遅延挿入

コネクション B の処理の進行がコネクション A の処理より進み過ぎて同期の品質を超過した場合、同期機構はコネクション A の処理が再開されるまでコネクション B の処理へのデータ供給を停止する。このようにコネクション B の論理時間の進行をデータ供給が再開されるまで停止して同期のずれを修正する処理を遅延挿入と呼ぶ。

我々は、メディア同期として実時間同期、イベント駆動型同期という 2 種類の同期の形態に着目した [10]。

実時間での同期として、テレビ電話における口の動きと声との同期、マルチメディアサーバのデータ再生における背景音、効果音、映像、テロップ間の同期等が考えられる。実時間での同期の様子を図 4-3(a) に示す。

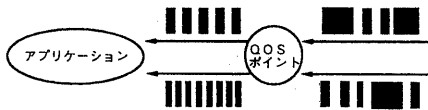


図 4-3(a): データ流量に基づく実時間同期

メディアが QOS ポイントに到達するまでは、なんら時間の制約はないが、一旦 QOS ポイントに到達すると、それぞれの単一メディアの品質が保証され、連続的にアプリケーションに渡されてゆく。

次に例えばマルチメディア情報のデータベースであるマルチメディアデータ概念モデルの提案では、マルチメディアプレゼンテーションのデータベース化などを対象に、「一かたまりの連続メディアデータ」を単位として、それらの前後関係、重なり関係などを時区間 (time interval) に基づいた関係で表現している。これは実時間同期とは別のイベント駆動的な同期の必要性を指摘したもので、連続/非連続メディアの統合が可能になると考えた。図 4-3(b) にイベント駆動型の同期の様子を示す。

これらをコネクションの優先度と、データ流量だけに基づいて実時間で同期させた場合、初期状態での同期を別途制御する必要があるたり、またシーン切替の瞬間に一時的にシーンを構成する音や映像といった各メディアを、優先度とは無関係にバリア同期させるような手法が必要となる。図 4-3(b) の白抜きのデータを using QOS ポイントでバリア同期が取られる。

この問題に対して、表現メディアの交換において必要な時に付随する情報を別途イベントとして交換し、その解釈によってデータ供給のタイミングを制御することが考えられる。

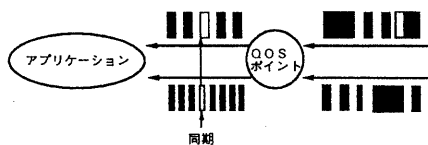


図 4-3(b): イベント駆動型同期

図 4-3(c) に、連続メディアと非連続メディアの同期の様子を示す。非連続メディアである、例えば、キーボード入力がある QOS ポイントに届くまで、連続メディアは停止している。

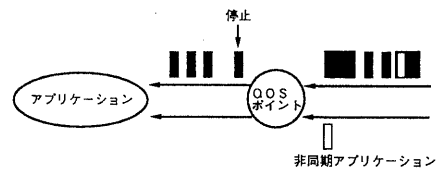


図 4-3(c): 非連続メディアとの同期

PDE-II の QOS ポイントにおける同期機構を図 4-4 に示す。同期機構は、同期管理プロセス、データ管理プロセスおよびバッファで構成される。単一メディアの品質はデータ管理プロセスが、またメディア間同期の品質は、同期管理プロセスによって保証される。同期機構の実装には次に説明する周期スレッドを用いる。

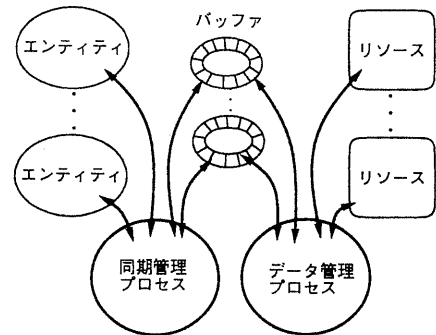


図 4-4: 同期機構

4.3 周期スレッド

PDE-II では、QOS ポイントを、エンティティが持つコネクションごとのデータの出入口であるエンドポイントとは別のコネクション上に独立に存在するデータの出入口とみなしたインタフェースを提供する。

また、QOS ポイント自身は、相手エンティティ側の QOS ポイントとの間で通信し、またはファイル/メモリ/デバイスなどの操作を行ない、かつ連続性の監視や同期の修正を行なう自律的なプロセスとして動作する。図 4-5 に示すように、固有のバッファ領域と動作プログラム、タイマを持つ周期スレッドによる実現を提案する [10]。この周期スレッドを用いて前述の同期機構の同期管理プロセスの実現を行なう。

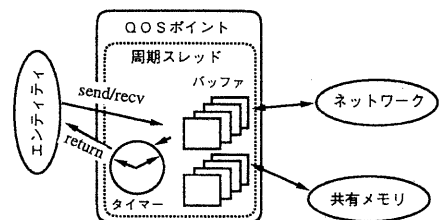


図 4-5: 周期スレッドを用いた同期機構

周期スレッドは以下の形式で呼び出される。我々は周期スレッドのインタフェースをメッセージパッシングの標準案 [6] に基づいて設計した。

$$\begin{bmatrix} \sqcup \\ a \\ v \end{bmatrix} \begin{bmatrix} p \\ b \\ n \end{bmatrix} - \begin{bmatrix} send \\ recv \end{bmatrix}$$

上の組合せは関数名を表しており、例えば、“ab_send()” のような関数が用意される。それぞれの動作を以下に説明する。

読み出し/書き込み

send エンティティから QOS ポイントへのデータの書き込み

recv エンティティの QOS ポイントからのデータの読み出し

読み出し/書き込み時のデータ構造

□ 指定なし。バッファに順番に書き込む。

a (array)

配列指定。同じ型、サイズのデータを等間隔 (引数指定) に取って、バッファに書き込む。データは全てメディアデータである。

v (variable)

任意のサイズ指定。任意の構造 (引数指定) に従って、バッファに書き込む。データは構造化されているので、メディアデータの他にメディア情報を付加することができる。

実時間同期

p (periodic)

処理のサイクル終了時刻にのみリターンする。その時点で、必要なデータ量に対して供給し又は供給されたデータの割合、他のコネクションとの同期の程度を QOS として返す。

なお、呼び出しの有無に関わらずサイクルは進むものとする。

b (block)

通常のソケットインタフェースと同様、サイクル時間とは無関係にスレッドの処理完了までリターンしない。

通信の場合は相手からの ack を待つ。

n (non-block)

サイクル時間とは無関係に、呼び出しの直後にリターンする。

4.4 エンティティ管理

PDE-II のエンティティは、エンティティ ID と呼ばれる一つのシステムリソース内で固有の ID 番号によって識別される [12]。エンティティ ID は、コネクション接続先のエンティティの指定などに用いる。エンティティ ID は、エンティティの実体であるプロセスの ID とプロセスを実行中のホストの ID で構成されるため、ネットワーク上で一意にエンティティを識別することができる。また、エンドポイントは特定のコネクションを指定する。各エンドポイントは、エンティティ内で固有の ID 番号をつけて管理される。

エンティティ間でコネクションを接続する場合、他のエンティティからのコネクション接続要求を待つエンティティは、接続要求を行なうエンティティに対して自らの存在を提示する必要がある。このための機構として PDE-II ではネームサーバを用意する。接続要求を待つエンティティは、ネームサーバに接続要求を待っているエンドポイントの ID と自らのエンティティ ID に対応する名前を登録することによって、自らのエンドポイントを他のエンティティに対して公開する。接続要求を行なうエンティティは、ネームサーバに登録された名前を問い合わせることによって接続先のエンティティ ID とエンドポイントの ID を知ることができる。

図 4-6 に示すようにネームサーバは各ホスト毎に一つ用意され、それぞれが各計算機内に存在するエンティティのエンドポイントの名前登録/解決の処理を行なう。

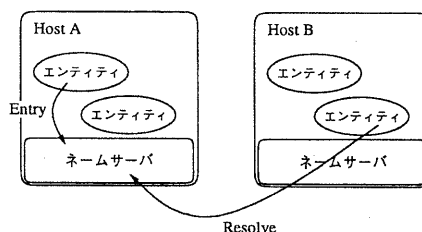


図 4-6: エンティティ管理機構

4.5 アプリケーション

我々が提案するマルチメディア処理モデルに基づいたアプリケーションの例として、図 4-7 に音声/画像会話ツールの処理モデルを示す [12]。

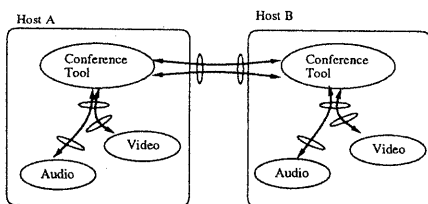


図 4-7: PDE-II 上でのアプリケーション

以下にアプリケーション上のエンティティ間でコネクションが張られ、データのやりとりが行なわれる様子を説明する。

1. エンティティが起動されると自分のエンドポイントを他に公開するためネームサーバに名前登録を行なった後、他のエンティティからのコネクション接続を待つ。
2. 他のエンティティに接続要求を行なうエンティティは、接続要求を待っているエンティティのエンティティ ID とエンドポイントの ID を知るためにネームサーバに問い合わせを行なう。
3. ネームサーバへに問い合わせが成功すると、接続したいエンティティに対してコネクションの接続要求を行なう。この時同時にコネクションの QOS に関する指定も行なう。
接続要求を待っているエンティティが接続要求を受けるとエンティティ間にコネクションが成立する。
4. エンティティ間の通信は、コネクションに対して読み書きをする関数で行なう。データの読み書きの QOS はコネクション管理機構および QOS ポイント管理機構によって保証される。

図 4-7 の音声/画像会話ツールは、以上述べた処理を基本として作成できる。従来、表現メディアの QOS の保証は、アプリケーションレベルでプログラマが行なう必要があり実現が困難であった。しかし PDE-II では QOS の保証をオペレーティングシステムが他のアプリケーションやユーザと調停しながら行なっているためプログラマの負担も軽減する。

5 おわりに

現在、我々は、リソース管理機構、メディア同期機構、周期スレッドおよびエンティティ管理機構といった基礎的な部分の設計、試作に取り組んでいる。リソース管理機構と周期スレッド機構は従来のオペレーティ

ングシステムのプロセス管理機構に、メディア同期機構とエンティティ管理機構は、従来のデバイス I/O 管理機構に組み込んでゆく予定である。

今後のハードウェアの性能の向上によって、単位時間当たりに処理できる絶対的な処理量の増加は期待できる。しかし、例えば、本稿で取り上げたマルチメディア処理における連続性の実現のために単一コネクションの品質である単位時間当たりの処理量や、メディア間の同期をハードウェアが保証することは困難である。このように、サービスの品質をいかにオペレーティングシステムに行なわせるかという問題はハードウェアの処理性能に拘らず、今後も重要な問題である。

参考文献

- [1] S. Casner and S. Deering, "First IETF Internet Audiocast", *ACM SIGCOMM Computer Communication Review*, pp.92-97, 1992.
- [2] H. Tokuda, Y. Tobe, S.T.-C. Chou and J. M. F. Noura, "Continuous Media Communication with Dynamic QOS Control Using ARTS with and FDDI Network," *In Proceedings of ACM SIGCOMM '92*, vol.22, no.4, 88-98, Oct. 1992.
- [3] C. W. Mercer, S. Savage and H. Tokuda, "Processor Capacity Reserver for Multimedia Operating Systems", *CMU-CS-93-157*, May. 1993.
- [4] "Experimental Internet Stream Protocol, Version 2 (ST-II)", *RFC1190*, Oct. 1990.
- [5] D. P. Anderson and G. Homsy, "A Continuous Media I/O Server and Its Synchronization Mechanism", *IEEE Computer*, vol.24, no.10, pp.51-57, Oct. 1991.
- [6] W. Gropp and E. Lusk, "A test implementation of the MPI draft message-passing standard", *TR ANL-92/47*, Argonne National Laboratory, Dec. 1992.
- [7] 安田, "マルチメディア符号化の国際標準", 丸善株式会社, 1991.
- [8] 岡村、吉川、稲垣、荒木: "QoS 指定可能なマルチメディアモデルの提案", 情報処理学会マルチメディア通信と分散処理ワークショップ 1993.
- [9] 稲垣、岡村、荒木: PDE-II におけるメディア間同期機構の実現に対する考察, 第 48 回 情報処理学会全国大会, 1H-6 1994.
- [10] 吉川、岡村、荒木: 周期スレッドを用いたマルチメディアデータの同期処理, 情報処理学会マルチメディア通信と分散処理研究会, 94-DSP-64-1 1994.
- [11] 松尾、岡村、荒木、福田: マルチメディア処理における OS レベルでのリソース・リザーベーション, 情報処理学会システムソフトウェアとオペレーティング・システム研究会, 94-OS-63-10 1994.
- [12] 田中、岡村、荒木: PDE-II におけるマルチメディアアプリケーションの実装について, 第 48 回 情報処理学会全国大会, 1H-9 1994.
- [13] 岡村、田中、荒木: QOS に基づいた電子メディアツールの評価, 情報処理学会マルチメディア通信と分散処理研究会, 94-DSP-64-2 1994.