

マルチメディア端末における適応的映像フロー制御方式

酒井 靖夫[†] 大久保 英嗣^{††}

IMT-2000 に代表される広帯域の伝送路を持つ無線通信環境では、従来よりもサイズが大きく動きがなめらかな映像表示が可能なモバイル・マルチメディア通信が可能になってくると考えられる。このような環境では、端末性能、動作中のアプリケーションプログラム、ネットワークの種類および通信状態、ユーザごとの利用形態が動的に変わる。このような動的変化に対してシステムが自律的に適応するためには、ミドルウェアレベルにおいてモニタリング、リソースサービス、CPU スケジューリング、フロー制御の 4 つの機能が必要となる。本論文では、フロー制御に着目し、受信映像のストリームの再生において、スレッドとバッファ構成の異なる再生方式を状況に合わせて適宜切り換え実行することにより、リソースを制御するための新しい手法を提案する。さらに、提案手法の実験による評価結果を示し、その有効性について議論する。

An Adaptive Video Flow Control Method for Multimedia Terminals

YASUO SAKAI[†] and EIJI OKUBO^{††}

In a wireless telecommunications environment with broadband transmission paths represented by IMT-2000, it is conceivable that mobile multimedia communications can display images with larger sizes than before and with smooth movements. In such an environment, the terminal performance, online application program, type and communication conditions of the network, and usage pattern of each user change dynamically. With such dynamic changes, monitoring, resource service, CPU scheduling, and flow control at the middleware level are necessary functions for the system to be adaptable from the viewpoint of autonomy. In this paper, attention is focused on flow control, and a new method is proposed to control resources by appropriately executing various methods for the thread and buffer configurations in accordance with the current conditions, in the play of received image streams. In addition, evaluation results obtained in experiments employing the proposed method are shown and discussions are made on the validity of the method.

1. はじめに

近年、マルチメディア通信、モバイル通信、パーソナル通信が急速に普及してきている。今後、歩行速度程度の移動環境において 384 Kbps の性能要求条件を持つ IMT-2000 といった広帯域の無線通信網が整備され、CIF (Common Intermediate Format) サイズ (360 画素 × 288 画素) の映像が、10 フレーム/秒で伝送可能になる。すなわち、現在の PHS を利用した移動テレビ電話と比較すると、サイズが大きく動きがなめらかな映像表示が可能なモバイル・マルチメディア通信が可能になってくると考えられる。このような

環境で連続メディアを通信する AP (アプリケーションプログラム) が動作する端末では、次のような環境が動的に変わる。

- ユーザごとの利用形態
ユーザの映像や音声に対する好みなど
- 端末性能
メモリ容量、CPU 処理速度、電源レベルなど
- 動作中の AP
AP の種類、動作状況、AP の数など
- ネットワークの種類
無線 LAN、IMT-2000 など
- 通信状態
伝送帯域、エラーレートなど

そのため、動的に変化する動作環境に即して、システムが自律的に AP の動作状態をユーザにとって望ましいように調整する機能が必要になると考えられる。

本論文では、マルチメディア端末としてテレビ会議

[†] 株式会社 ATR 環境適応通信研究所
ATR Adaptive Communications Research Laboratories

^{††} 立命館大学理工学部情報学科
Department of Computer Science, Faculty of Science
and Engineering, Ritsumeikan University

などの専用のシステムではなく、複数のアプリケーションプログラムが動作する携帯型モバイル・マルチメディア端末を想定している。端末には、連続メディアストリームを扱うマルチメディア通信 AP (以降、MM-AP) のほかに、ワープロ、表計算、ドローイングなどの各種 AP (以降、非 MM-AP) がリソース (メモリ、CPU 時間) が許す限り同時に動作可能な状態にある。映像を受信中に、別の資料を見る必要が発生したとする。現在のリソース使用状況では、リソースが少なく新たに資料を開くことができない場合、何らかの方法によりリソースを解放し、資料を開く非 MM-AP を動作可能にする必要がある。この場合の解決方法として、次の 2 つの QoS 制御方式が考えられる。1 つは、送信側との交渉によってリソース使用量の少ない新しいアプリケーション QoS (AP-QoS: 映像メディアの場合はフレームレート、画像サイズ、画像品質など、音声メディアの場合はサンプリングレート、圧縮方法など) に変更し、リソースを解放する方式である。これに関する研究として、end-to-end の QoS 保証を目的とした QoS アーキテクチャの研究^{1),2)}、QoS 制御にエージェント技術を導入する研究などがある^{3),4)}。もう 1 つは、受信端末において再生されるストリームに対するユーザの要求に基づきメディア再生品質を変更し、ストリーム処理に使用するリソースの使用量の割当てや配分を制御することによって対応する方式である。受信端末側で独自にリソース制御を行うことは、モバイル・マルチキャスト通信においてモバイル端末と通信する基地局が QoS を変更する機能を持たず送信側の AP-QoS を 1 つの端末のために容易に変更できない場合や、端末が自律的にリソースを解放もしくは獲得することにより変化への応答性を高めたい場合に必須となる技術である。これに関しては、楽観的 CPU 予約に基づく QoS 制御機構⁵⁾、端末上の複数ストリームの資源管理を実現する「QoS チケット」モデル⁶⁾などの研究がある。しかし、これらの研究は再生される連続メディアの品質を評価し、その評価値を QoS 制御に用いたものではない。

本論文では、フロー制御を端末において連続メディアを受信し再生するまでのデータ処理の流れと捉え、モバイル・マルチメディア端末において受信映像再生処理時に、適応的に端末のリソース制御を実現する新しいフロー制御方式を提案する。本フロー制御方式では、再生される映像の質とリソース使用量の異なる複数の再生方式の中から現在の状況に最適な再生方式を動的に選択実行する枠組みを提供する。これまで再生品質に関する研究として、再生品質の定義式からユー

ザが満足する最低限の品質を保証する QoS 仕様を決定する研究⁷⁾などがある。本論文では、再生品質の評価を連続メディア再生時に動的に再生方式を切り換える手がかりとして用いる。再生時に使用されるメモリと CPU の使用量をそれぞれメモリ使用評価値、CPU 使用評価値として数値化し、再生される映像の質も再生満足度として数値化する。これらの数値化されたメモリ使用評価値、CPU 使用評価値、再生満足度から計算した適応度を比較することによって、現在の状況に最適な再生方式を選択し実行することが可能になる。

以下、本論文では、2 章でマルチメディア端末のソフトウェア構成とミドルウェアの機能についてまとめる。次に、3 章でミドルウェアの機能であるフロー制御に関して、モバイル・マルチメディア環境を想定した適応的フロー制御方式 (AFC: Adaptive Flow Control) について述べる。最後に、4 章で AFC を用いた実験結果を示し、その有効性について議論する。

2. マルチメディア端末

本章では、提案するフロー制御方式である AFC が動作するマルチメディア端末の構成とその動作について述べる。

2.1 マルチメディア端末の全体構成

ユーザの操作、ネットワーク通信状況、リソース使用状況の変化に対して AP (MM-AP および非 MM-AP) を適応的に動作させるマルチメディア端末のソフトウェアは、OS、ミドルウェア、AP による 3 階層で構成する (図 1 参照)。ミドルウェアには、提案するフロー制御機能のほかに、モニタ、スケジューラ、リソースサーバの機能が必要になる。新規 MM-AP 起動時の AP、ミドルウェア、OS の 3 者間の制御の流れは次のようになる。ユーザは MM-AP を起動する。起動された MM-AP は、ユーザの要求を取り込み、リソースサーバ、端末内にすでに実行されている他の MM-AP、通信相手端末の MM-AP と通信を行い、AP-QoS を決定する。MM-AP は、リソースサーバに対して決定された AP-QoS を通知する。さらに、MM-AP は、フロー制御へ AP-QoS、CPU リソース、メモリリソース使用パラメータを通知し、通信セッションを開始する。以下に、各々の役割と機能について述べる。

(1) AP

AP には、ミドルウェアと協調して連続メディアストリームを扱う MM-AP と、ワープロ、表計算ソフトのように文字や図などの離散メディアを扱う非 MM-AP が複数混在している。特に、MM-AP は、環境の変化

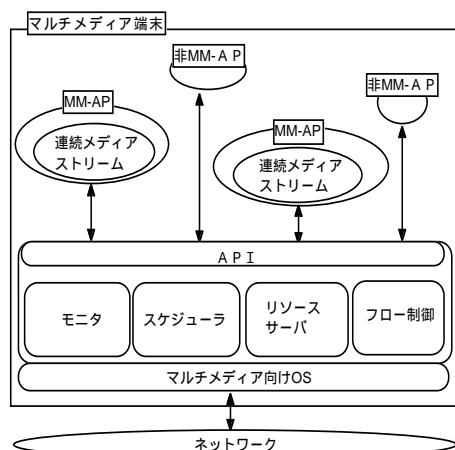


図1 マルチメディア端末のソフトウェア構成

Fig. 1 Software configuration of a multimedia terminal.

を検知し、その変化に対して意志決定し、行動する機能を備えている必要がある。そのためには、ミドルウェアとの通信、相手端末のMM-APとの交渉、AP-QoSの決定、ユーザの好みの獲得などの機能を実現しなければならない。

(2) モニタ

モニタは、各種リソースの使用状況やメディア処理状況などをモニタリングする。モニタリングの対象としては、以下のものがある。

- システム全体のCPU使用量とストリームごとのCPU使用量
- システム全体のメモリ使用量とストリームごとのメモリ使用量
- 受信エラーレート（無線通信におけるBit error rate）
- モバイル端末の電源レベル
- ストリームごとのデコード単位のデータ受信状況
- ストリームごとのデコード単位のデータ再生状況

(3) スケジューラ

リソースの状況に変化が生じた際、AP-QoSを変更しなければならない場合がある。その場合は、MM-APは、新しいAP-QoSを決定するために、リソースサーバ、端末内の他のMM-APと通信し、相手端末のMM-APと通信をしなければならない。その際、メディア処理よりも通信処理の優先度を高くして、CPU時間を優先的に割り当てることにより、変化への即応性を高める。

(4) リソースサーバ

リソースサーバには、次の2つの機能が必要になる。

● リソース情報提供サービス

MM-APは、AP-QoSを決定する際、要求するAP-QoSが現在のリソース状況で実現可能か否かを判定しなければならない。そのために、MM-APに対して、現在使用可能なAP-QoSに関する情報を提供する機能が必要となる。CPU処理速度や搭載しているメモリ容量などの端末の性能が異なる場合、同じAP-QoSでもそれを実行するために必要となるリソース量は異なってくる。MM-APが端末の性能まで考慮に入れてAP-QoSを決定する場合、MM-APを性能の異なる端末で動作させるために、AP-QoSと各種リソース使用量（CPU使用時間、メモリ使用量、伝送データ量）との間のマッピングを調整する必要がある。ミドルウェアにAP-QoSと各種リソース使用量との間のマッピング機能を持たせることにより、動作させる機種への依存性がなくなり、MM-APの独立性が向上する。

● リソース調整起点サービス

何らかのAPが、リソースの獲得を要求した際、それがエラーで終了したときもしくはエラーで終了しそうなどとき、リソースサーバは動作中のMM-APへその旨通知する。たとえば、非MM-APが、mallocをコールしたがメモリが確保できなかったとする。エラー情報は、当初mallocをコールした非MM-APには通知せずに、動作中のMM-APへ通知する。通知を受けたMM-APは、通信相手端末のMM-APと交渉しAP-QoSを劣化させメモリを解放するか、後述のフロー制御において再生品質を劣化させメモリを解放するかのいずれかの方法によって、メモリリソースを解放する（どちらの方策をとるかはMM-APの作り方に依存する）。MM-APは、リソースサーバに対してリソースを解放したことを報告する。報告を受けたリソースサーバは、再度mallocを実行する。この機能に関しては、仮想メモリ環境を想定したメモリ予約機能の研究⁸⁾などがある。

(5) フロー制御

マルチメディア端末において、受信した映像ストリームを再生する際、MM-APの要求（きれいな映像が見たいのか、映像の質よりリソースを節約したいのかなど）や受信状況（遅延、ジッタなど）に合わせて、動的に映像品質と再生に使用されるリソースを制御する機能が必要となる。本論文では、この手法について述べている。

3. AFC (適応的フロー制御)

3.1 AFCの概要

モバイル通信環境において動作するMM-APにとって、そのMM-APが使用可能な端末のリソースであるメモリ、CPU使用時間(モバイル端末では、バッテリー寿命を延ばすために、CPU使用時間はできるだけ少なく抑える必要がある)は、固定端末と比較して制限を受ける。さらに、無線通信では以下の理由により、有線通信よりもジッタの発生確率およびジッタ時間が大きくなると考えられる。

- 電波状態の悪化にともなうエラーレートの増加により再送が多発する。
- FEC (Forward Error Correcting) を付加しエラー耐性制御を施すことにより、伝送速度が低下する。

そのため、モバイル通信環境において受信した連続メディアのリアルタイム再生は、有線通信とは異なり、ある程度の通信状態に起因する再生の乱れは避けられないと考えられる。本論文で提案するフロー制御は、連続メディア再生品質を追求することが目的ではない。モバイル端末のように有限のリソース状態において連続メディアを再生する際、ユーザの要求に応じてリソース使用量と再生品質を適応的に調整することを目的としている。

本論文で提案するAFCの特徴は、以下のとおりである(図2参照)。

- 再生品質と再生時に使用されるリソース使用量が異なる3種類の連続メディア再生方式(以降、再生方式)を、フレームジッタ(再生映像1画面分のデータ到着ジッタ)に応じて、動的に切り換える枠組みを提供する。
- 再生方式の選択は、各再生方式が独自に、再生時の満足度とそのときに費やされるリソース使用量の評価値を計算し、その値を比較することによって行う。

したがって、再生方式の選択は、再生方式に依存する再生品質とその処理のために使用されるリソースとのトレードオフになると考え、次式(1)の評価によって行う。

$$\begin{aligned} \text{再生適応度} &= \text{再生満足度} - \text{リソース使用評価値} \\ &= \text{再生満足度} \\ &\quad - (\text{メモリ使用評価値} + \text{CPU使用評価値}) \end{aligned} \quad (1)$$

各再生方式に関して、独自に式(1)を計算し、再生適

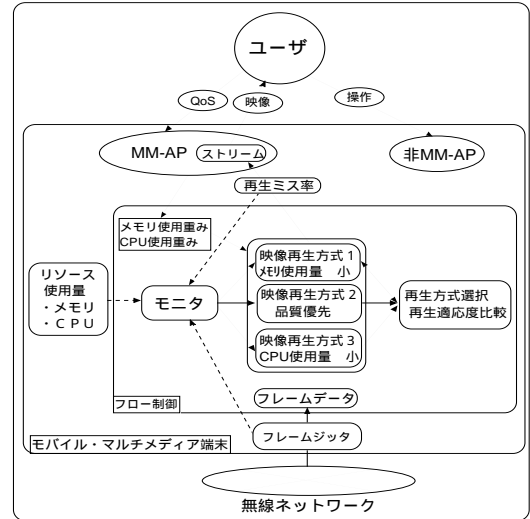


図2 AFCの概念図

Fig. 2 Processing flow of AFC.

応度を算出する。算出した再生適応度の中で最も大きい値を持つ再生方式を選択する。すなわち、現在の状況において、妥当なリソース量を使用してできる限り良い映像を再生するのに適していると判定し、ストリームの再生方式として選択し実行する。式(1)において、リソース使用評価値とともに再生満足度を導入することによって、同じリソース使用評価値であっても、ユーザにとって満足度の高い再生方式が選択されるようにしている。リソースとしては、メモリとCPU使用時間を考えている。再生満足度、メモリ使用評価値、CPU使用評価値については、3.2節以降で詳述する。

さらに、AFCでは、各ストリームごとに独立に式(1)を計算することによって、ストリームごとに異なる再生方式を選択することを可能にしている。たとえば、2種類の映像ストリームがあった場合、ストリーム1の再生はメモリの使用量の少ない再生方式により処理し、ストリーム2は品質重視の再生方式により再生することが可能になる。

3.2 再生満足度

式(1)中の再生満足度は、次式(2)により与えられ、再生される連続メディアに対するユーザの満足度を表す。再生満足度は、フレーム数/秒、画像サイズ、画像品質、再生ミス率に影響されると仮定している。

$$\begin{aligned} \text{再生満足度} = \text{関数}(\text{フレーム数/秒}, \\ \text{画像サイズ}, \\ \text{画像品質}, \\ \text{再生ミス率}) \end{aligned} \quad (2)$$

表 1 再生満足度テーブル
Table 1 Satisfactoriness table.

再生ミス率 (%)	0	10	20	30	40	50	60	70	80	90
再生満足度 (5 fps)	80	75	60	50	30	15	10	8	6	5
再生満足度 (10 fps)	100	95	75	30	20	10	6	5	2	0

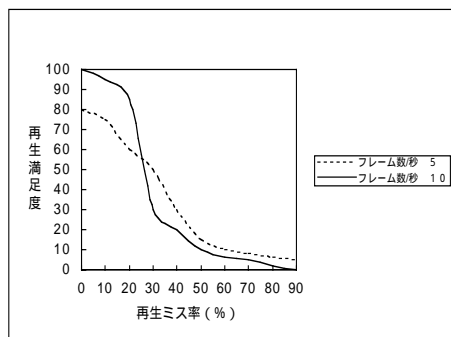


図 3 再生満足度曲線

Fig. 3 Satisfactoriness curves of playing video.

- フレーム数/秒
1 秒間に表示されるフレーム数・枚数が多いほど満足度は高く、少ないほど満足度は低くなる。
- 画像サイズ
画像の縦横の画素数・画素数の多い画像ほど満足度は高い。画素数の少ない画像の満足度は低い。
- 画像品質
画像コーデックの品質制御パラメータ。再生される画像の質に影響を与える。きれいな画像ほど満足度は高くなる。
- 再生ミス率
連続メディアが欠落もしくは遅延により周期内に実行されない割合。この割合が低いほどなめらかな映像が再生されるので満足度が高くなる。

式 (2) では、再生満足度が大きいほど、ユーザの再生映像への満足度は高いとする。特に、画像サイズと画像品質が一定の場合、その再生満足度はフレーム数/秒と再生ミス率の影響のみを受け、次式 (3) で与えられる。実験では、式 (3) を使用している。

$$\text{再生満足度} = \text{関数}(\text{フレーム数/秒}, \text{再生ミス率}) \quad (3)$$

表 1 と図 3 に、画像サイズと画像品質を一定にしたとき、フレーム数/秒 (fps) が 10 と 5 の場合の各々の再生ミス率の変化にともなう再生満足度の変化を示す。再生映像の質の評価は、ユーザの主観評価によるところが大きく、また個人個人によっても評価が異なるので、再生満足度テーブル (表 1) は、映像メディアを見るユーザが設定するとした。表 1 では、フレーム数/秒が 10 と 5 の最高満足度 (再生ミス率 0 のと

き) はそれぞれ 100 と 80 としている。この値は、フレーム数/秒が 10 と 5 では、10 の場合の方がユーザが好ましく思っていることを示す。

また、再生ミス率の計算において、再生方式によっては、フレームジッタが直接再生ミス率に結びつく場合もあれば、逆にフレームジッタに影響されず再生ミス率が低く保たれる場合もある。さらに、再生ミス率は、フレームジッタからある程度予測のできるものと考えている。選択され実行されている再生方式の再生満足度の計算には、モニタリングされた再生ミス率の実測値を使用する。選択されていない再生方式の場合は、モニタリングされたフレームジッタの実測値から予測される再生ミス率を使用して再生満足度を計算する。

3.3 メモリ使用評価値

式 (1) 中のメモリ使用評価値は、式 (4) と (5) によって与えられる。

$$\begin{aligned} \text{メモリ使用評価値} &= \text{単位メモリ使用価値} \\ &\quad \times \text{再生処理に必要なメモリ使用量} \quad (4) \\ \text{単位メモリ使用価値} &= \text{再生満足度の最大値} \\ &\quad \times \text{メモリ使用重み} \\ &\quad \div \text{最大メモリ使用量} \quad (5) \end{aligned}$$

以下に、式中の各々の項目について説明する。

- 単位メモリ使用価値
式 (5) で計算される単位メモリ使用価値は、単位あたりのメモリリソースを使用することに対する評価値である。単位メモリ使用価値は、式 (4) によって計算されるメモリ使用評価値と再生満足度とのスケーリングの機能を兼ねている。単位メモリとして、1K バイトや 1 画像分などを指定する。
 - 再生満足度の最大値
再生満足度テーブル (表 1 参照) 中の最大値であり、100 である。
 - メモリ使用重み
MM-AP が AFC に対して設定する値である。メモリリソースを使用することに対する重みと捉える。
 - 最大メモリ使用量
すべての再生方式のメモリ使用量の中の最大値である。

式 (5) の各々のうち、再生満足度の最大値と最大メモリ使用量は、通信セッションの開始時には決まった値となる。したがって、式 (5) の評価は、メモリ使用重みの値により動的に変動する。メモ

リソースに対する制約が少なく、ストリームの受信再生のために大きなメモリリソースを割当て可能な場合、メモリ使用重みを 0 に近い値に設定する。逆に、メモリリソースを使用することに制約を加えたい場合、このメモリ使用重みとして大きな値（実験では 1.0）を設定する。

- 再生処理に必要なメモリ使用量

式 (4) 中の再生処理に必要なメモリ使用量は、再生方式に依存する。メモリ使用評価値を算出するうえで、各再生方式のメモリ使用量の差を反映することが重要である。各再生方式のコードサイズの差は、各再生方式が映像データを保持するために必要とする画像バッファサイズの差と比較するとごく小さい。したがって、再生処理に必要なメモリ使用量として画像バッファサイズを使用する。

- 選択されている再生方式の再生処理に必要なメモリ使用量

連続メディアの受信再生処理のために実際に使用されている画像バッファサイズ

- 選択されていない再生方式の再生処理に必要なメモリ使用量

モニタリングされたフレームジッタの実測値から予測される再生処理に必要となる画像バッファサイズ

- メモリ使用評価値

式 (4) によって与えられるメモリ使用評価値は、式 (1) の再生適応度を計算する際、再生満足度からの減算要因となっている。メモリ使用評価値が小さい場合、映像の再生においてメモリリソースの影響が小さいことを表す。逆に、メモリ使用評価値が大きい場合は、影響が大きいことを表す。品質が劣化している状況で、ユーザがきれいな再生映像を望む場合やメモリリソースが十分ある場合は、メモリリソースを多く使用して映像を再生する。この場合、式 (5) にあるメモリ使用重みとして 0 に近い値を設定することにより、単位メモリ使用評価値は 0 に近い値となる。さらに、単位メモリ使用評価値の値が小さくなると、式 (4) のメモリ使用評価値も 0 に近い値となる。これは、式 (1) の適応度を評価する際に、メモリリソースの使用量の影響が少ないことを示しており、メモリを多く使用する再生方式が選択されやすくなることを意味する。逆に映像の質よりもメモリリソースを解放したい場合もしくはメモリリソースが少ない場合は、メモリ使用重みとして 1 に近い値を設定する。これにより、単位メモリ使用評価値が大きく

なり、式 (4) のメモリ使用評価値も大きな値となる。その結果、メモリ使用量の少ない再生方式が選択されやすくなる。

3.4 CPU 使用評価値

式 (1) 中の CPU 使用評価値は、メモリ使用評価値と同様に式 (6) と (7) によって与えられる。

CPU 使用評価値

$$= \text{単位 CPU 使用評価値} \times \text{再生処理に必要な CPU 使用量} \quad (6)$$

単位 CPU 使用評価値

$$= \text{再生満足度の最大値} \times \text{CPU 使用重み} \div \text{最大 CPU 使用時間} \quad (7)$$

以下に、式中の各々の項目について説明する。

- 単位 CPU 使用評価値

式 (7) で計算される単位 CPU 使用評価値は、単位時間 (1 秒) あたりの CPU リソースを使用することに対する評価値である。単位 CPU 使用評価値は、式 (6) によって計算される CPU 使用評価値と再生満足度とのスケーリングの機能を兼ねている。

- 再生満足度の最大値

再生満足度テーブル (表 1 参照) 中の最大値であり、100 である。

- CPU 使用重み

MM-AP が AFC に対して設定する値である。CPU リソースを使用することに対する重みと捉える。

- 最大 CPU 使用時間

すべての再生方式中の単位時間あたりの CPU 使用時間の最大値である。

式 (7) の各項のうち、再生満足度の最大値と最大 CPU 使用時間は、通信セッションの開始時には決まった値となる。したがって、式 (7) の評価は、CPU 使用重みの値により動的に変動する。この CPU 使用重みは、メモリ使用重みと同様に、CPU リソース制約が低く、ストリームの受信再生に十分な CPU 時間が割り当てられていることが可能な場合は、0 に近い値が設定される。逆に、CPU リソースに制約がある場合は、相対的に大きな値 (実験では 1.0) が設定される。

- 再生処理に必要な CPU 使用量

式 (6) の再生処理に必要な CPU 使用量は、再生方式に依存する。

- 選択されている再生方式の再生処理に必要な CPU 使用量

連続メディアの受信再生処理のために実際に

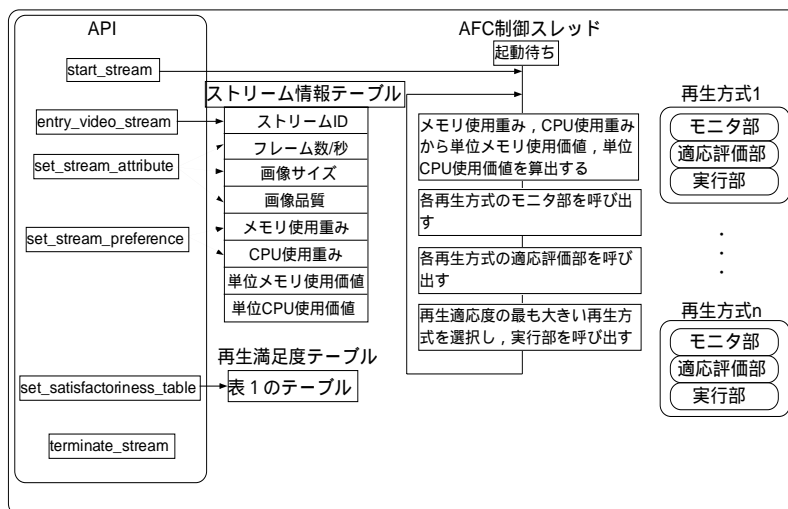


図4 AFCのAPIと内部構造
Fig. 4 API and internal structure of AFC.

使用されている CPU 使用時間を用いる。

- 選択されていない再生方式の再生処理に必要な CPU 使用量

再生方式の CPU 使用時間の大部分は、デコードをソフトウェアで行っているので、1 枚の画像のデコードと表示に費やされる。したがって、モニタリングされたフレームジッタの実測値から予測される再生フレーム数/秒と 1 枚の画像のデコード時間および表示時間の積として必要 CPU 使用時間を推定して用いる。

● CPU 使用評価値

式 (6) によって与えられる CPU 使用評価値は、メモリ使用評価値と同様に、式 (1) の再生適応度を計算する際、再生満足度からの減算要因となっている。CPU 使用評価値が小さい場合、映像の再生において CPU リソースの影響が小さいことを表す。逆に、CPU 使用評価値が大きい場合は、影響が大きいことを表す。

ユーザがきれいな再生映像を望む場合、CPU リソースが十分確保できる場合、バッテリーが十分な場合は、CPU 時間を長く使用して映像を再生する。この場合、式 (7) にある CPU 使用重みとして 0 に近い値を設定することにより、単位 CPU 使用価値は 0 に近い値となる。さらに、単位 CPU 使用価値の値が小さくなると、式 (6) の CPU 使用評価値も 0 に近い値となる。これは、式 (1) の適応度を評価する際に、CPU リソースの使用量の影響が少ないことを示しており、CPU を多く使用する再生方式が選択されやすくなることを意

表2 AFCのAPI
Table 2 API of AFC.

API	機能
entry_video_stream ()	新規ストリーム再生の登録
set_stream_attribute ()	ストリームの AP-QoS (フレーム数/秒, 画像サイズ, 画像品質) の設定
set_stream_preference ()	ストリームのメモリ使用重み, CPU 使用重みの設定
start_stream ()	ストリーム再生処理の開始
stop_stream ()	ストリーム再生処理の中断
terminate_stream ()	ストリーム再生処理の終了
set_satisfactoriness_table ()	再生満足度テーブルの変更

味する。逆に映像の質よりも CPU リソースを解放したい場合もしくは CPU リソースが少ない場合は、CPU 使用重みとして 1 に近い値を設定する。これにより、単位 CPU 使用価値が大きくなり、式 (6) の CPU 使用評価値も大きな値となる。その結果、CPU 使用量の少ない再生方式が選択されやすくなる。

3.5 プログラムインタフェース

AFC では、表 2 に示す API (Application Program Interface) を用意している。この API による AFC の処理の大まかな流れは次のようになる。図 4 に API と、API により設定される AFC 内部データとの関連を示す。

- (1) set_satisfactoriness_table により再生満足度テーブルを AFC 内部に通知する。
- (2) MM-AP は、entry_video_stream により再生ストリームを定義する。
- (3) set_stream_attribute により AP-QoS を設定

表 3 再生方式のソフトウェア構成

Table 3 Software configuration of each method.

ブロック	機能
モニタ部	適応性を評価するために必要なメモリ使用量, CPU 使用量, フレームジッタ, 再生ミス率などのデータを収集する.
適応評価部	モニタ部により収集されたデータを基に, 式 (1) を計算し再生適応度を求め, 現在の状況に対する適応性を自己評価する.
実行部	映像メディアの受信再生を実行するためのスレッドの起動とメモリの獲得, 終了処理のためのスレッドの消滅とメモリの解放を実行する.

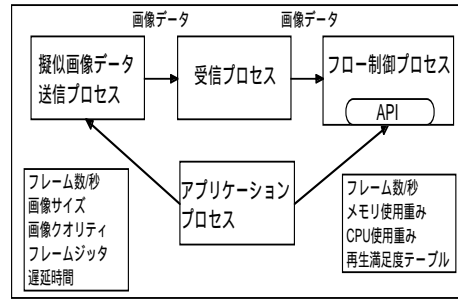


図 5 ソフトウェア構成

Fig. 5 Software configuration for experiments.

する.

- (4) `set_stream_preference`により,メモリ使用量, CPU 使用量を設定する.
- (5) `start_stream`により,ストリーム再生を開始する.
- (6) ストリーム再生中,`set_stream_preference`により,メモリ使用量, CPU 使用量を変更する.これにより,選択される再生方式を切り換えることが可能となる.

3.6 AFC の処理手順

AFC 内部における処理は,次の3つのステップから構成される.

ステップ 1 再生方式の準備

フレームジッタによる再生映像の劣化度合いや受信再生処理時のリソース使用量が異なる複数の再生方式を準備する.各再生方式は,図4に示す3つのプログラムで構成する.その機能を表3に示す.

ステップ 2 AFC サーバの起動

図4に示すAFC制御スレッドを起動する.

ステップ 3 受信再生セッション

AFC制御スレッドは,以下のように動作する(図4参照).

- (1) APIを介して設定されたメモリ使用量とCPU使用量から,単位メモリ使用量と単位CPU使用量を計算する.
- (2) AFC制御スレッドから各再生方式のモニタ部を呼び出し,現在実行されている再生方式が消費するメモリ使用量とCPU使用時間をモニタリングする.
- (3) AFC制御スレッドにより,すべての再生方式の適応評価部を呼び出し,各再生方式が算出した再生適応度を比較し,再生適応度の最も大きい再生方式を選択する.
- (4) 選択した再生方式が現在実行されている再生方式と異なる場合は,現在実行している再生方式を終了し,選択された再生方式の実行を開始する.

以上の処理を繰り返すことによって,AFCは実行される.受信再生セッション中の動作をストリームの数だけストリームごとに独立に実行する.AFCでは,状況の変化への適応性を各再生方式が自己評価した値の大小によって判定する.その結果,再生方式の追加と,それにもなうAFC制御スレッドにおいて方式を選択する閾値の調整などの方式切替調整を行う必要がなくなる.

4. 実験による評価

映像メディアを受信再生する3種類の方式を装備したAFCを実装し,その性能を検証した.実験環境は以下のとおりである.

- 使用機種: エプソン社製 Endeavor Pro-330L
- CPU: Celeron 300 MHz
- メモリ: 32 MBytes
- OS: RealTime-Mach (Version MKNG008)

実験に使用したソフトウェアの構成を図5に示す.擬似画像データ送信プロセスは,送信端末とネットワークをシミュレーションし,アプリケーションプロセスからフレーム数/秒,画像サイズ,画像品質,フレームジッタ,遅延時間を受け取る.擬似画像データ送信プロセスは,メモリ上に保持するJPEG圧縮した画像データを,指定されたフレーム数/秒,指定されたフレームジッタ,遅延時間に従い受信プロセスへ送り出す.アプリケーションプロセスから表2に示すAPIを介して,フロー制御プロセスを制御し,フレーム数/秒,メモリ使用量, CPU 使用量,再生満足度テーブルをフロー制御プロセスへ通知する.フロー制御プロセスは,受信タイミングにともない受信プロセスから画像データを読み出し,AFCを用いて再生する.AFCにおける再生は,JPEGのデコードまでを実行している.

4.1 映像メディア再生方式の概要

再生方式は,図6,図7,図8に示される方式1,2,

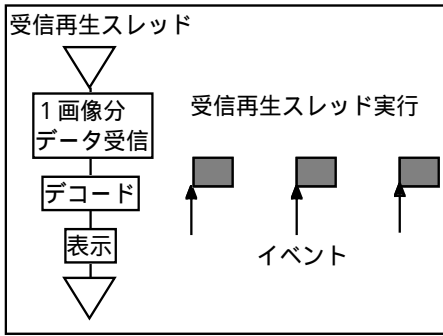


図 6 映像再生方式 1

Fig. 6 First method for playing video.

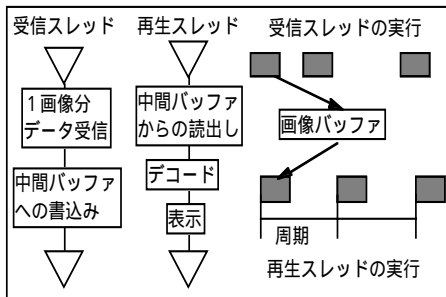


図 7 映像再生方式 2

Fig. 7 Second method for playing video.

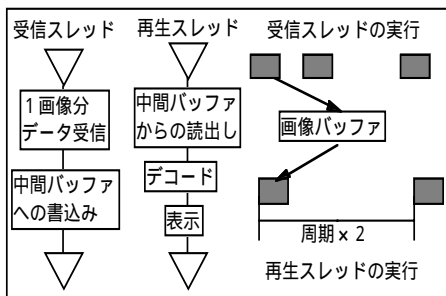


図 8 映像再生方式 3

Fig. 8 Third method for playing video.

3の3種類である。各方式の処理概要は、以下のとおりである。なお、各方式の特徴を表4にまとめた。

- 再生方式 1
1画面分の画像データの受信、デコード、表示の一連の処理を、1つのデータ到着時に起動されるイベント駆動型のスレッドで実行することにより受信映像を再生する。
- 再生方式 2
データ到着時に起動される1画面分の画像データの受信、画像バッファへの書き込みを行うイベント駆動型の受信スレッド、および画像バッファからの読み出し、デコード、表示を周期的に実行

表 4 再生方式の特性比較
Table 4 Comparison of three methods.

再生方式	再生方式 1	再生方式 2	再生方式 3
メリット	メモリ使用量が他の2方式より少ない。	フレームジッタを画像バッファで吸収するため再生映像の乱れが少ない。	CPU使用量が他の2方式より少なく、フレームジッタによる再生映像の乱れが少ない。
デメリット	フレームジッタにより容易に再生映像が乱れる。	メモリ使用量が方式1より多い。	メモリ使用量が方式1より多く、再生品質が方式2より悪い。

する再生スレッドの2種類のスレッドと画像バッファによって構成される。

- 再生方式 3
再生方式2と同じスレッド、バッファ構成を持つ。再生スレッドの実行周期が方式2の2倍になっている点異なる

4.2 各再生方式による再生適応値の計算

- 再生満足度の導出
再生満足度は、3.2節に示した式(3)を使用して導出する。式(3)のフレーム数/秒には、再生方式1と2ではアプリケーションプロセスから通知されたフレーム数/秒の値を代入し、再生方式3ではアプリケーションプロセスから通知されたフレーム数/秒の1/2の値を代入し再生満足度を導出する。すべての再生方式において、実行中は再生ミス率を代入し再生満足度を導出する。再生方式1の場合、待機中は再生ミス率の代わりにフレームジッタ値を代入する。再生方式2と3の場合、待機中の再生ミス率の値として、これら方式の特性上フレームジッタに影響されず一定の値0を代入する。すべてのコードとデータが実メモリに存在する携帯型モバイル端末の動作を想定しているため、コードおよびデータのページフォルトの発生頻度はごく少ないとし、再生満足度への影響は無視する。
- 単位メモリ使用価値の計算
実験では、1画面分の圧縮データを固定サイズと考え、これをメモリ使用評価値計算のメモリ単位とした。再生方式1では1単位、他の2種類の方式では1秒分の画像を保持することとし、フレーム数/秒に等しい単位のメモリを使用する。再生満足度の最大値は、表1から100となる。したがって、式(5)で与えられる単位メモリ使用価値

は次のように計算される．

単位メモリ使用価値

$$= \text{再生満足度の最大値} \\ \times \text{メモリ使用重み} \\ \div \text{最大メモリ使用量} \\ = 100 \times \text{メモリ使用重み} \\ \div \text{フレーム数/秒}$$

● 単位 CPU 使用価値の計算

1 画面分の圧縮データの受信および画像バッファへの登録、読み出し時間は、画像のデコード表示時間と比較すると非常に短時間なので実験では無視した．したがって、1 画面分の圧縮データをデコード表示する時間を CPU 使用評価値計算の CPU 単位とした．再生方式 1, 2 ではフレーム数/秒に等しい CPU 単位を消費し、再生方式 3 はフレーム数/秒 \div 2 の CPU 単位時間を消費する．再生満足度の最大値は、表 1 から 100 となる．したがって、式 (7) で与えられる単位 CPU 使用価値は次のように計算される．

単位 CPU 使用価値

$$= \text{再生満足度の最大値} \\ \times \text{CPU 使用重み} \\ \div \text{最大 CPU 使用時間} \\ = 100 \times \text{CPU 使用重み} \\ \div \text{フレーム数/秒}$$

4.3 実 装

AFC と API の RT-Mach への実装について述べる．図 4 に示される AFC 制御スレッドは、start_stream を呼び出すことにより、RT-Mach のプリミティブ rt_thread_create を使用して周期スレッドとして起動される．図 7, 8 に示される再生方式 2 および再生方式 3 の再生スレッドは、AFC 制御スレッドから方式の実行部が呼び出されたときに、同様に rt_thread_create を使用して周期スレッドとして生成する．スレッドのスケジューリングポリシーは、rate monotonic を使用している．また各方式の画像バッファの獲得は、malloc により獲得される．各モニタ量の計測方法を、表 5 に示す．

4.4 条 件

実験では各種パラメータを表 6 のように設定し、メモリ使用重み、CPU 使用重み、フレームジッタの 3 種類のパラメータを変化させ、選択される再生方式の変化を観測した．

4.5 結 果

メモリ使用重みと CPU 使用重みをともに低く（ともに 0.0）設定したときの実験結果を図 9 に示す．実験

表 5 モニタ方法

Table 5 Monitoring methods.

モニタ項目	手 法
メモリ使用量	各再生方式によって確保された画像バッファ量．
CPU 使用量	再生終了時刻と再生開始時刻の差を単位時間あたり合算し集計したもの．
再生ミス率	1 枚の画像の処理終了時刻と前画像の処理時刻との差と再生周期を比較することによって行う．単位時間あたりのミスの割合を計算する．画像の抜けも再生ミスとして扱う．
フレームジッタ	1 枚の画像の受信時刻と前画像の受信時刻との差と再生周期を比較することによって行う．単位時間あたりのフレーム遅れの割合を計算する．

表 6 実験条件

Table 6 Conditions for experiments.

フレーム数/秒	10 フレーム/秒
遅延時間	60 ミリ秒
1 画像のデータサイズ	10 Kbytes
画像サイズ	320 画素 \times 240 画素
1 画像のデコード表示時間	約 10 ミリ秒
再生方式 2, 3 の画像バッファ量	10 画面分
メモリ使用重み	0.0 ~ 1.0 まで 0.2 刻み
CPU 使用重み	0.0 ~ 1.0 まで 0.2 刻み
フレームジッタ	0 ~ 90 まで 10 刻み
AFC 制御スレッドの周期	3 秒

結果から、フレームジッタの増加にともない再生方式 1 から再生方式 2 へ再生方式が変わっていることが読み取れる．これは、リソース使用量よりも再生品質が優先され、再生品質が最良となる方式がフレームジッタの増加に合わせて選択実行されていることを示す．

メモリ使用重みの値を高く（1.0）設定したときの実験結果を図 10 に示す．メモリ使用重みの値を高く設定すると、フレームジッタの増加とは無関係にメモリ使用量の少ない再生方式 1 が選択実行されることが分かる．

CPU 使用重みの値を高く（1.0）設定したときの実験結果を図 11 に示す．CPU 使用重みの値を高く設定すると、CPU 使用量の少ない再生方式 3 がフレームジッタの割合に関係なく選択実行されることが分かる．

メモリ使用重みと CPU 使用重みをともに高く（ともに 1.0）設定したときの実験結果を図 12 に示す．メモリ使用重みと CPU 使用重みをともに高く設定したとき、フレームジッタの増加にともない再生方式 1 から再生方式 3 へ再生方式が変わっていることが読み取れる．これは、再生品質よりもリソース使用量が少ない再生方式が、フレームジッタの増加に合わせて選択実行されていることを示す．

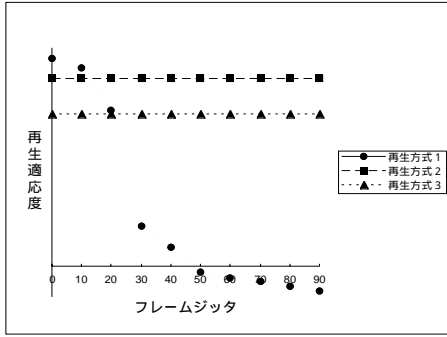


図 9 メモリと CPU リソースともに制約の少ない場合の再生適応度 (メモリ使用重み 0.0, CPU 使用重み 0.0)

Fig. 9 Value of adaptability when the restriction for memory and CPU is low.

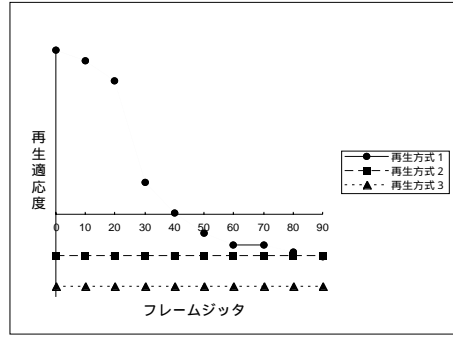


図 11 メモリリソースの制約が大きい場合の再生適応度 (メモリ使用重み 1.0, CPU 使用重み 0.0)

Fig. 11 Value of adaptability when the restriction for memory is low.

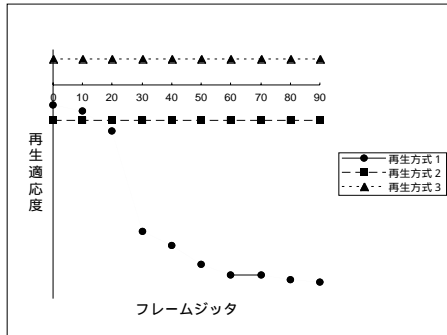


図 10 CPU リソースの制約が大きい場合の再生適応度 (メモリ使用重み 0.0, CPU 使用重み 1.0)

Fig. 10 Value of adaptability when the restriction for CPU is high.

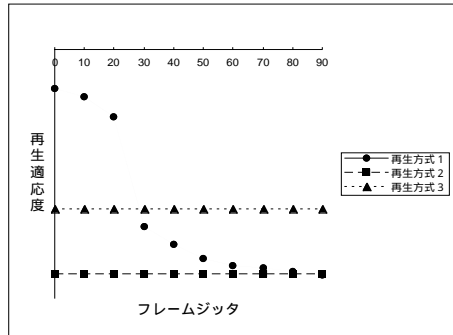


図 12 メモリと CPU リソースともに制約が大きい場合の再生適応度 (メモリ使用重み 1.0, CPU 使用重み 1.0)

Fig. 12 Value of adaptability when the restriction for memory and CPU is high.

実験結果から、AFC が意図したとおりに動作していることが確認された。このように、AFC の上位層から CPU 使用重みおよびメモリ使用重みの両パラメータを与え、適応度評価式 (1) を用いて、メリット/デメリットを持った複数のストリーム再生方式を選択的に実行するフロー制御方法は、再生されるストリームに対する上位層の要求を反映することを可能にする。

本実験では、motionJPEG 方式の映像を受信再生する 3 つの再生方式に対して AFC を適用し、その特性を検証した。AFC は、処理に必要なメモリと CPU 使用時間のリソース量およびフレームジッタによるメディア再生品質への影響を見積もることが可能な再生方式に対して有効と考えられる。

5. おわりに

通信状況、リソース使用状況、ユーザのニーズなどの環境変化に対して動的に適応するマルチメディア端末には、ミドルウェアレベルにおいて、モニタ、リソ

スサーバ、スケジューラ、フロー制御の機能が必要になる。本論文では、特に、フロー制御のための新しい手法である AFC を提案し、RT-Mach を使用したパーソナルコンピュータ上で動作実験を行い、その有効性を確認した。メモリ使用量、CPU 使用量、再生される映像の満足度が異なる再生方式を切り換える制御構造は有効と考えられる。

今回は、コーデックに前画像に影響されない motion-JPEG 方式を採用し、通信環境の変化としてフレームジッタの影響のみを考慮している。今後は、モバイル環境を想定したコーデックで、しかも前画像との差分により映像を圧縮する MPEG4 方式を使用して AFC の性能を評価する必要がある。さらに、フレームジッタだけでなく遅延時間の再生満足度に対する影響も考慮した AFC を実現していく予定である。

謝辞 本研究の機会ならびに貴重なご助言をいただいた株式会社 ATR 環境適応通信研究所の小宮山社長に感謝の意を表する。

参 考 文 献

- 1) Aurrecochea, C., Campbell, A.T. and Hauw, L.: A Survey of QoS Architectures, *Multimedia Systems*, Vol.6, No.3, pp.138-151, ACM (1998).
- 2) Nakajima, T. and Tezuka, H.: A Continuous Media Application Supporting Dynamic QoS Control on Real-Time Mach, *Multimedia*, pp.289-297, ACM (1994).
- 3) 中沢 実, 須田飛志, 酒井宏三, 服部進実: エンドユーザワーキング - ユーザアダプティブエージェントによる QoS 制御, 信学技報, CQ97-6, pp.39-46 (1997).
- 4) 小菅昌克, 山崎達也, 荻野長生, 松田 潤: マルチエージェントによる適応的 QoS 制御方式, 信学論(B), Vol.J82-B, No.5, pp.702-710 (1999).
- 5) Nakajima, T.: A Dynamic QoS Control Based on Optimistic Processor Reservation, *ICMCS'96*, pp.95-103, IEEE (1996).
- 6) Kawachiya, K. and Tokuda, H.: Dynamic QoS Control Based on the QOS-Ticket Model, *ICMCS'96*, pp.78-85, IEEE (1996).
- 7) Staehli, R., Walpole, J. and Maier, D.: *A quality-of-service specification for multimedia presentations*, pp.251-263, Springer-Verlag (1995).
- 8) Nakajima, T. and Tezuka, H.: Virtual Memory Management for Interactive Continuous Media Applications, *ICMCS'97*, pp.415-423, IEEE (1997).

(平成 11 年 12 月 24 日受付)

(平成 12 年 7 月 5 日採録)



酒井 靖夫

昭和 33 年生。昭和 57 年京都大学工学部情報工学科卒業。同年三洋電機(株)入社。日本語ワードプロセッサのソフトウェア開発および PDC, CDMA 方式携帯電話端末の開発に従事。平成 9 年 ATR 環境適応通信研究所出向。モバイル環境における連続メディア制御方式を研究, 現在に至る。電子情報通信学会会員。



大久保英嗣(正会員)

昭和 26 年生。昭和 52 年北海道大学大学院工学研究科情報工学専攻修士課程修了。同年(株)日立製作所ソフトウェア工場入所。主として FORTRAN コンパイラの開発に従事。昭和 54 年京都大学工学部情報工学科助手。昭和 60 年同講師。昭和 62 年同助教授。平成 3 年立命館大学理工学部情報学科教授となり, 現在に至る。工学博士。オペレーティングシステム, データベースシステム, 分散システム, 実時間システム等の研究に従事。著書に「オペレーティングシステムの基礎」(単著, サイエンス社)、「オペレーティングシステム」(編著, オーム社)等がある。電子情報通信学会, 日本ソフトウェア科学会, システム制御情報学会, ACM, IEEE-CS 各会員。