

会話型操作を伴う分散マルチメディアシステムにおける同期機構

大野 隆一, 相田 仁, 齊藤 忠夫

東京大学工学部

本論文では, Ethernet でつながれた複数の UNIX ワークステーション上に現在構築中の分散マルチメディアシステム (DMSIC) において負荷変動に柔軟に対応しつつ情報提示の際のメディア間の同期を維持する機構を示す. 本同期機構では, 各情報源からのメディア送出は基準となる時間軸に同期させることで, 各情報提示ノードでの情報提示は理想的な再生点に合わせることで同期を維持する.

本機構の評価を行なった結果, 本機構を用いることで各ノードが自ノードの CPU の余裕度に応じて情報送出 (提示) を行ないながら各ノードでの情報提示の際のメディア間同期を取ることができることを確認した.

キーワード 分散マルチメディアシステム, 同期機構, 会話型操作

A Synchronization Mechanism in a Distributed Multimedia System with Interactive Control

Ryuichi Ohno, Hitoshi Aida, and Tadao Saito

Faculty of Engineering, the University of Tokyo

In this paper, we propose a technique to keep multiple *Retrieved Media Streams* (RMS) synchronized at the time of presentation while absorbing load variations in a *Distributed Multimedia System with Interactive Control* (DMSIC) which has been implemented on UNIX workstations connected by Ethernet. An evaluation of this technique shows that each node can adjust the delivery (or presentation) of media units according to the capacity of the CPU without being affected by a node with a lower capacity CPU.

keywords distributed multimedia system, synchronization mechanism, interactive control

1 まえがき

近年、ネットワーク技術及びコンピュータ技術の発展に伴い、ネットワークを介してマルチメディア情報を送ることで会議、教育などの様々な活動を支援するシステムの研究・開発が盛んに行なわれてきている。

このようなシステムにおいては様々な情報源から送られてくる複数のメディア情報を各ノードで提示する際にメディア間の同期を取ることが必要となる^[1]。

従来、ビデオ・オン・デマンドのように一つの蓄積型装置から取り出されたマルチメディア情報を一つのノードで提示するシステムにおいてはメディア間同期について様々な検討がなされてきたが、多地点間の会議や教育を支援するシステムのように共通の情報が複数のノードにおいてほぼ同時に提示されるようなシステムにおいてはメディア間同期の問題はあまり問題にされてこなかった。

我々はこのようなシステムにおいてメディア間の同期を取る手法についての検討、及び、システムの試作を行なっている。

本論文では、現在試作中のシステムにおいて負荷変動に柔軟に対応しつつ情報提示時にメディア間の同期を取る機構を示す。論文では、まず、システム概念、及び、現在試作中のシステムの概要について述べる。次に、試作中のシステムにおいて付加変動を吸収しながら情報提示の際のメディア間の同期を維持する手法について述べる。最後に、本同期方式の性能評価について述べる。

2 システムの概要

本節ではシステムの概要について述べる。

2.1 システム概念

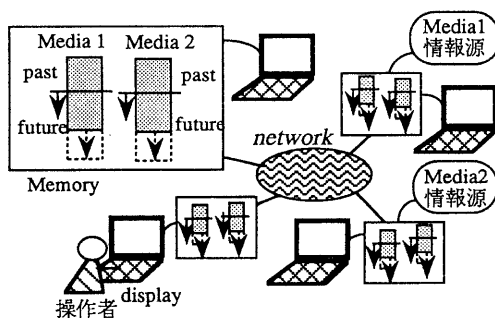


図 1: システム概念

本システム (図 1) では、複数の情報源から別々の通信路 (コネクション) を介してマルチキャストされてくる複数の (意味的につながりのある) メディア情報が複数のノードにおいてほぼ同時に提示される。

各ノードには display などのマルチメディア表示用の装置、及び、メディア間のずれの吸収などに用いられるメモリを備え、さらにその内幾つかのノードにはメディア情報の入った Hard Disk などの情報発生源となる装置を備えている。そして、この内一つのノードの利用者のみが操作者となり再生、巻き戻しなどの会話的な操作を行なう。

2.2 試作システムの概要

前節で述べたシステムを Ethernet(10Mbps) で結ばれた複数台の UNIX ワークステーション上に実装した。

試作システムではシステム中に一つ存在し、初期化などの際に用いられる管理プロセスと各参加ノードごとに起動される動作プロセスの 2 種類のプロセスを用いる。また、これらのプロセス間に各メディア情報及び制御情報ごとに connection を張ることで情報をやりとりする。情報送出、表示などは各動作プロセス中で定期的にスケジュールされる procedure を用いて行なう。

情報源としてはハードディスク中に格納された動画像 (1 フレーム 21,600 bytes)、テキスト情報 (1 つにつき 20 bytes)、及び音声 (8000 bytes/sec) の 3 つを用い、操作命令としては再生、一時停止、逆再生、早送り、巻き戻し、高速再生、高速逆再生などの命令をサポートしている。

本システムの基本的な動作を以下に示す。

操作者 操作者となったノードではユーザからのイベントを監視し、イベントがあった際にはそのイベント種別とイベントが発生した時間 (フレーム番号) を全ての参加者ノードに通知する。

情報源 情報源となる装置を備えるノードでは操作者ノードからの指示に応じて各参加者にフレーム番号付きのメディア情報を送る。

情報提示 情報提示を行なうノードでは情報源から送られてきたメディア情報を一旦メディアごとに設けられているバッファに蓄える。情報提示ノードではさらに、操作者ノードからの指示に応じてメディアバッファ上の情報を適宜ユーザに提示する。

主なメディア操作の概要は以下のようなになる。

再生 再生命令と再生開始位置からなる命令バケットを各ノードに送る。各ノードでは、通知された再生開始位置から再生を開始する。

一時停止： 停止命令と停止位置からなる命令パケットを各ノードに送る。各ノードでは、通知された停止位置の画面を表示し、停止する。

巻き戻し： 巻き戻し命令を各ノードに送る。巻き戻し中は操作者ノード以外で画面が表示される必要はない。

試作システムではネットワークの伝送遅延を予測できない。従って、各ノードではバッファ中に提示できるメディア情報が無くなった時点でメディア情報の到着を待つ。

また、本システムでは「全ての参加者ノードで少なくとも操作者ノードで提示されるメディア情報と同じだけのメディア情報が提示される。」こととする。

これを満たすためにはパケット、操作のための情報、及びバッファリングされている情報に時間情報を付加する必要がある、ここではこの時間情報としてフレーム番号を用いる。

3 同期方式

本節では試作システムにおける同期方式について述べる。

3.1 概要

本システムのように QOS を指定することで資源を確保することができないネットワークと端末により構成されるシステムの場合、他のプロセスによる影響を排除できない。従って、このような環境でメディア間の同期を取るには各ノードでの負荷変動、遅延時間変動などを吸収するための何らかの機構が必要となる。

メディア間同期の手法^[3, 2]としては、基準となる時間軸に全てのメディアの進行を合わせる方法、基準となるメディアの進行に他のメディアの進行を合わせる方法などが考えられる。

本システムでこれらの方法を用いる場合、各ノードでのバッファ上の再生位置を意図的に変えることができない。つまり、メディアバッファを有効に利用することができなくなる。

そこで、本システムでは、各ノードのバッファ中に理想的な再生点を定め、この理想的な再生点に各メディアの再生点を近付ける方法を用いている。

本システムでは再生間隔の基準となる時間 (T_{int}) は一定とする。そして、情報提示の際には各メディアの再生間隔ごとに適切なフレームを選択、提示することでバッファ上の再生位置を意図的に変化させる。

また、本システムでは情報源には予め同期付けられた情報が格納されていることを想定しているため、Acme^[2]

のように各フレームが送出される際に基準となるメディアに合わせてフレーム番号を付加する方法では予め意図していたような同期が取れない可能性がある。

そこで、本システムでは、各メディアフレームの送出を基準となる時間軸に合わせることにする。

以上述べてきたように、本システムでは各情報源からのメディア送出は基準となる時間 (T_{int}) に同期させることで、各情報提示ノードでの情報提示は理想的な再生点に合わせることでメディア間同期を維持する。

3.2 情報源ノードでの調整

3.2.1 video, text の場合

video, text の場合、情報送出を行なわない情報送出区間を設けることでメディア送出を基準となる時間軸に合わせる。

1. skip の算出

式 (1) により skip を算出する。ここで、 T_{int} は情報送出 procedure のスケジュール間隔、 T_{sch} は情報送出 procedure が実際にスケジュールされるのにかかった時間であり、前回送出したフレームより skip + 1 フレーム先のフレームを送出することで情報送出を基準となる時間軸に合わせるができる。

$$skip = [(T_{sch} - T_{int}) / T_{int}] \quad (1)$$

また、skip が大きいことは負荷が大きいことを示しており、負荷を下げるために何らかの工夫をすることが望ましい。本実装では skip + 1 先のフレームを送出した後、引き続き $[skip/2]$ 回のスケジューリングの際には情報送出を行なわない。

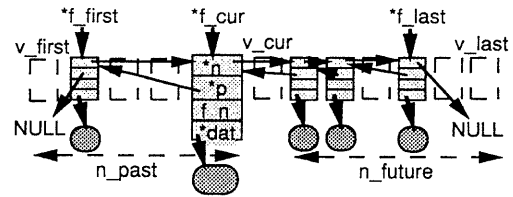


図 2: メディアバッファ

情報源から各ノードに送出するデータパケットのデータへの付加情報はフレーム番号と l_skip, r_skip (それぞれ skip, $[skip/2]$ のいずれかに対応) となる。

各情報提示ノードでは送られてきたデータをフレーム番号 f_num と l_skip, r_skip を参考にして各ノードのメディアバッファに格納する。メディアバッファの構造を図 2 に示す。

2. skipの平滑化

一時的な負荷の増大により *skip* が大きな値を取ることは情報提示の際の連続性を損なうため、望ましくない。本実装では情報送出の実時間からのずれを許し、*skip* の上限 *U_SKIP* と下限 *L_SKIP* を適応的に変化させることで *skip* の平滑化を図りつつ負荷変動に対応する。以下に、平滑化の手法を示す。

- *skip* > *U_SKIP* の場合、*skip* を *U_SKIP* に設定し実時間からの情報送出の遅れを示す変数 *delay* に (*skip* - *U_SKIP*) を加える。
- *skip* < *L_SKIP* で、*delay* > 0 の場合、*skip* を *L_SKIP* に設定し、*delay* から (*L_SKIP* - *skip*) を引く。
- *delay* の大きさに応じて *U_SKIP*、*L_SKIP* を調整する。

3. 高速再生、巻き戻しなどへの対応

高速再生などでは、通常再生よりも *N* 倍の速度で情報を送出する必要がある。本システムでは、*skip* を *N* 倍し、さらに *N* - 1 だけ増やすことで送出速度を *N* 倍にしている。

3.2.2 audio の場合

audio の場合、情報送出を行なわない情報送出区間を設けることは望ましくない。そこで、video, text の場合に加え、無音区間を利用することなどにより最悪の場合を除いて *skip* が起こらないようにしている。

1. 無音区間の利用

音声情報源では、無音部を送出する必要がないため、有音区間のみを次々と送出することで基準となる時間軸よりも情報送出を先行させることが可能となる。

- *delay* < *MAX_DELAY* ならば、*l_skip*, *r_skip* を 0 にする。そして、その分だけ *delay* を増やす。
- さらに *delat* < *MAX_ADVANCE* (< 0) ならば、*delay* をインクリメントし、情報送出を行なわない。
- 情報送出を行なう際には、有音区間までに存在する無音区間の数だけ *delay* を減らす。

音声情報源から各ノードに送出するデータパケットにはフレーム番号 *l_skip*, *r_skip* 以外に無音部の長さを付加する必要がある。

また、音声情報を格納するメディアバッファには図 2 の情報以外に、各フレームが有音か無音かを示す情報が必要となる。

2. M 倍送出

audio の 1 フレーム分の情報量は video などと比べてかなり小さいため、各情報送出区間において最大 *M* フレーム分の情報送出を許すこととする。この *M* の値を早送りの際などの倍速パラメータ *N* よりも大きく取っておけば、早送りの際などにも質の良い音声を提示できる。

以上述べてきた方法により、情報源からの情報送出はほぼ基準となる時間軸に追従する。

3.3 情報提示ノードでの調整

3.3.1 video, text の場合

式 (1) と同様の式により、情報提示 procedure がスケジュールされる度に *skip* を求める。*skip* が 0 より大きい場合、(*skip* + 1) 回再生点の更新を行なったフレームを提示する。また、引き続き [*skip*/2] 回の情報提示 procedure スケジュールの際には情報の提示は行わない。(つまり、[*skip*/2] 回同じ情報が提示され続ける。ただし、各スケジュールの際に再生点の更新は行う。)

そして、再生点の更新の度に、理想的な再生点との位置関係により *skip* または *pause* を行なうことで理想的な再生点に追従する。

また、メディアバッファ中に *v_current* のフレームが存在しない場合、メディアバッファ中のどの実フレームを選択するかの方針を以下に示す。

1. *v_current* に最も近い実フレーム *nearest_f* を選択。
2. *v_current* が 2 つの実フレーム *n_first*, *n_last* のちょうど真ん中にある場合、再生中とすると [*v_prev* < *n_first*] ならば *n_first* を [*v_prev* ≥ *n_first*] ならば *n_last* を選択。(逆再生中も同様。)

このような方法を用いることで情報提示ノードでは最適なフレームを負荷に応じて提示できる。

また *N* 倍速に関しては、*skip* 及び再生点の更新位置を *N* 倍先にすることで実現している。

3.3.2 audio の場合

音声の場合、video や text で用いた方法では *skip* の度に音が途切れてしまう。

そこで、音声の場合には、以下のようにして、音声データの queue を常にある程度満たしておき、*skip* の際にも queue が無くなって音が途切れることがないようにする。

- 音声提示 procedure のスケジュールの度に queue 中の音声データの大きさを示す変数 *v_queue* をスケジュール間隔に比例した大きさだけ減らす。そして、*v_queue* の大きさがある固定長 *s_queue* になるまで

前回のスケジュールの際に最後に挿入したフレーム lv_q の次のフレームから順次 queue を満たしていく。

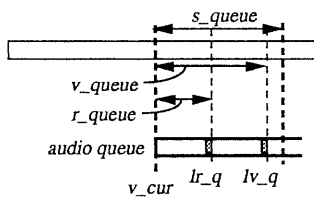


図 3: audio queue

- lv_q と v_queue により現在提示されているフレームを予測し $v_current$ とする。この $v_current$ と理想的な再生点との位置関係により $skip$, $pause$ を行なうかどうか決める。ただし、 $skip$, $pause$ は即座には行なわず、 lv_q 以降 $SEARCH$ フレーム分のフレームについて無音部を検査し、できるだけ無音部において（有音部において行なわれているはずの） $skip$, $pause$ を行なうようにする。つまり、音楽などのように有音部が続く場合以外は、無音部において $skip$, $pause$ を行なうことで有音部の音声品質を同期よりも優先させる。

$skip$, $pause$ は lv_q の次またはそれ以降に挿入されるフレームに適用されるため、 s_queue が大きいほど理想的な再生点への収束は遅れる。（付加変動による音の途切れを防ぐため、 s_queue はある程度以上小さくできない。）

- 音声データは 1 秒当たり 8000 バイトずつ再生されることになっているが、その再生速度のわずかなずれも、長時間再生を続ける場合、かなり大きな同期ずれを引き起こす可能性がある。そこで、本システムでは、無音区間が続く場合、実際には queue にデータを挿入せず、有音部が挿入される際にすでに挿入されているはずの無音部のデータを挿入してから有音部のデータを挿入することとし、無音部で queue が空になった際に同期をとり直すことができるようにしている。
- 本システムでは逆再生、高速逆再生の場合には、音声は提示しない。高速再生の場合には、queue に挿入するフレームを N ごとにするすることで N 倍速を実現する。また、早送り、巻き戻しの際には音声は通常で再生方向に提示される。これは、 N 倍速で動く仮想的な再生点と、実際の再生点の位

置関係により、実際の再生点を次々と有音部の開始位置に飛ばしていくことで実現している [6]。

3.4 情報提示ノードでの再生点のずれへの対応

本システムでは、“バッファ中の理想的な位置に再生点を持ってくる”ことで安定した（逆）再生状態を実現し、各ノードでのメディア間のずれも自ずと小さくなることによりメディア間の同期を取る。

3.4.1 理想的な再生点

本システムでは各ノードにおいて、各メディアバッファの共通部分の先頭から一定の距離にある位置を理想的な再生点としている。

3.4.2 再生点の調整

実際の再生点 (F_{real}) と理想的な再生点 (F_{ideal}) の位置関係に応じて $skip$, $pause$ を行なうことで F_{real} を F_{ideal} に近づける [4]。本システムでは F_{real} が F_{ideal} から遠いほど $skip$ や $pause$ の頻度を大きくしている。

4 同期機構の評価

3章で述べた同期機構を Ethernet で接続された 3 台の UNIX workstation を用いて評価した。

4.1 評価環境

3 台のマシンの役割を以下に示す。

SAIL (Sparc station 10) 情報源 (video, text), 情報提示の 2 つの役割を兼ねる。

IPX (Sparc station IPX) 情報源 (audio), 操作者, 情報提示の 3 つの役割を兼ねる。

SS1 (Sparc station 1) 情報提示のみを行なう。

バッファの大きさは video, text に対しては 20 フレーム分、audio に対しては 20+20 フレーム分とし、 $T_{int} = 100ms$ とした。また、以下の測定は全て他のプロセスが走っていない環境で行なった。（つまり、負荷変動の吸収機構はこの場合、各ノードの [CPU の能力 - 負荷] の差を吸収する機構として働くことになる。）

初期状態としては各メディアバッファが 0 フレーム目からのメディア情報で半分だけ満たされている状態とした。そして、IPX において再生命令が出され、各ノードで 0 フレーム目からの再生が行なわれた際に各ノードにおいて実際に各フレームが送出された時間と提示された時間を測定した。ただし、音声フレームの情報提示時

間はそのフレームが音声 queue に挿入された時間と、その時点での queue の長さにより算出した値を用いている。

4.2 評価結果

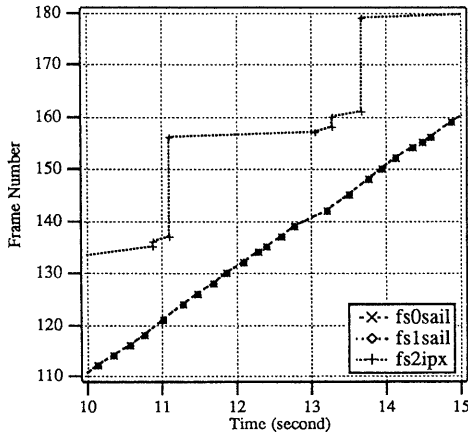


図 4: 情報源でのフレーム送出時間

評価結果を図 4, 及び, 図 5 に示す。

図 4 は情報提示開始 10 秒後から 5 秒間に情報源から送出されたフレームを, 図 5 は同じ時間に情報提示ノードで提示されたフレームを示している。

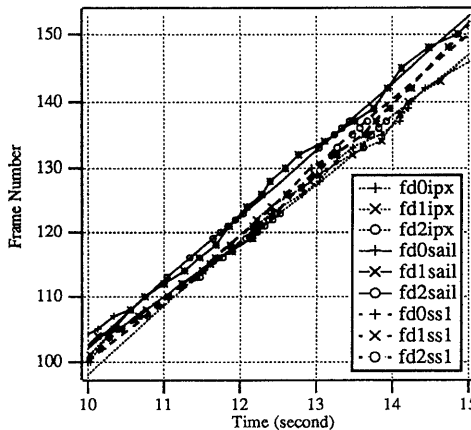


図 5: 情報提示ノードでのフレーム提示時間

4.3 考察

図 4 より以下のようなことが言える。

- video, text の送出時間はほぼ実時間に追従している。
- audio の送出は他メディアよりもかなり先行している。しかし, その先行は一定時間内に留まっている。

また, 図 5 からは以下のようなことが言える。

- 各ノードで 3 つのメディアはかなりよく同期している。
- SAIL, IPX, SS1 の順で提示されたフレーム数が多い。

以上の考察から, 3 章で述べた負荷変動を吸収する機構, 及び, メディア間の同期機構は本システムにおいて有効に動作していると考えられることができる。

5 むすび

本稿では, DMSIC において, 負荷変動に柔軟に対応しつつ情報提示の際のメディア間の同期を保つ機構を提案し, 試作システムを用いた評価によりその有効性を示した。

今後, ネットワーク, 端末などにおいても QOS を申告することで資源を確保できるようになると考えられるが, そのような環境においても本稿で提案した同期機構を用いることで QOS の申告に柔軟性を持たせることができ (必要な CPU 資源の上限と下限を申告するなど), システム中の資源を有効に活用することができる。

本システムの今後の課題としては, 新たなメディアの導入, 情報源に蓄えられているメディアと, 操作者の声などの live media との同期である。

参考文献

- [1] Ralf Steinmetz: "Synchronization Properties in Multimedia Systems", IEEE Journal on selected areas in communications, 8, 3, April 1990.
- [2] Anderson D.P., and Homys G.: "A Continuous Media I/O Server and Its Synchronization Mechanism", IEEE COMPUTER, 24, 10, pp.51-57(1991).
- [3] 藤川和利, 下條真司, 松浦敏雄, 西尾章治郎, 宮原秀夫: "分散型ハイバメディアシステム Harmony における情報同期機構の実現", 信学論 (D-I), J76-D-I, 9, (1993-9)
- [4] 大野, 相田, 齊藤: "会話型操作を伴う分散マルチメディアにおける情報同期機構の実現", 第 48 回情報処理学会全国大会, 6C-1, 1994
- [5] 矢野, 寺西, 大野, 相田, 齊藤: "会話型操作を伴うマルチメディアシステムにおける同期方式", 情報処理学会マルチメディアと分散処理研究会, 65-11, 1994.