

## マルチメディアの会話型操作における同期方式の検討

大野 隆一 寺西 祐人 相田 仁 齊藤 忠夫

東京大学 工学部

多様なメディアを一体として扱うマルチメディア通信では、メディア間の同期を取ることが必要となる。本稿では、各メディアがそれぞれ別々の通信路を通して複数のノードにマルチキャストされる場合にメディア間の同期を維持しつつ巻き戻しなどの会話的な操作を行なうための手法について検討し、そのためのプロトコルを示す。

### 1. はじめに

画像、及び音声、データなどのマルチメディア情報を伝送する方式に関する研究が盛んに行なわれてきている[1,2]。

マルチメディア通信を行なう一つの方式としてすべてのメディア情報を単一データストリームを介して送る方式がある。このような方式はCATVネットワークのような放送型のネットワークなどで使われている。

一方、それぞれのメディアをメディアごとに別々のデータストリームを介して送るという方式もある。次世代ネットワークとして期待される広帯域ISDN(B-ISDN)では、その伝送方式を非同期転送モード(ATM)としているが、ATMでは各メディアを別々のコネクションを通して送ることで各メディアはそれぞれのメディアに応じたサービス品質を指定することができる。また、マルチメディア高次通信においては通信する両者の間に自動翻訳などの様々な支援サービスを挿入することが考えられるが、この場合にも、各メディアごとに独立のコネクションを確立し、異なる経路で異なる処理を行なう。

このように各メディアを別々のコネクションを通して伝送することでメディアの特性に応じ

た効率的な通信が可能になり、メディアごとの情報処理なども容易になる。しかし、各メディアを別々のコネクションを通して送ることによりメディア間の時間的ずれは必然的に大きくなり、各ノードでメディア間の同期をあわせることが必要となる[3,4]。

本稿では、ネットワークで結合された分散システム上で、複数のメディア情報が複数の情報源から各々別々の通信路(又はコネクション)を通して複数のノードにマルチキャストされる場合に、メディア間の同期を維持しつつ巻き戻しなどの会話的な操作を行なうための手法について検討し、そのためのプロトコルを示す。

まず、2章では簡単なシステムイメージと、メディア操作の概要について述べる。3章ではメディアを簡単に分類する。4章ではメディア間の同期を取るためにパケットなどに付加する情報について検討する。5章ではメディア操作のためのプロトコルを示す。6章ではPacket Formatについて触れ、最後に端末構成を示す。

### 2. システムの概念

#### 2. 1. システムの概念

システムの概念は図1のように各ノードにdis-

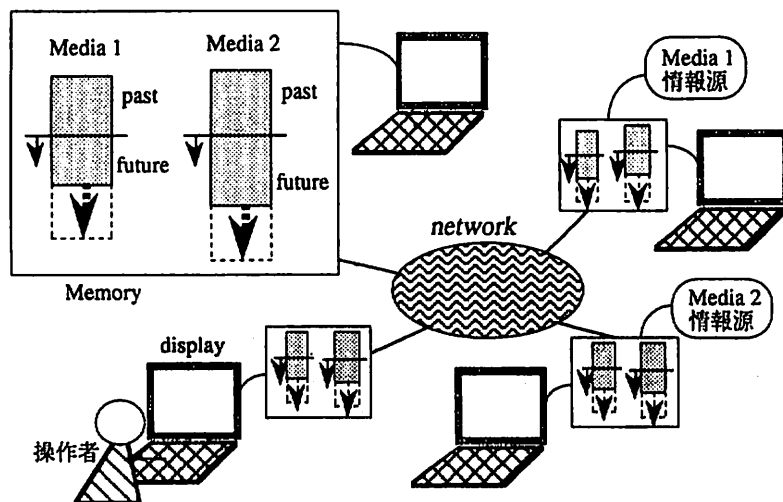


図 1：システムイメージ

playなどのマルチメディア表示用の装置、及び、メディア間のずれを吸収するためのメモリを備え、さらにその内幾つかのノードにはvideo装置などの情報発生源となる装置を備えている、といったものとする。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. メディア操作の概要

主なメディア操作の概要について以下に示す。但し、操作を行なえる人は一時に一人に限るとし、操作権がノード間を移動するといった方法で操作権の委譲が行なわれるとする。また、このような操作を実現するためのプロトコルについては後で述べるが、ここではとりあえず操作コマンドはすべてのノードにブロードキャストされ、現在を中心としてある程度の期間の情報が各ノードでバッファリングされているとする。

[一時停止] ノード内で各メディアは同期して再生されるが、それぞれのノードでは微妙に時間のずれた場面が再生されていると考えられ、また、一時停止コマンドも通信遅延を伴って伝送される。従って、すべてのノードでのメ

ディアの再生を同一の箇所ではめるためには、一時停止コマンドと一緒に停止箇所を示す情報を送る必要がある（このあたりのプロトコルについては後で述べる。）。この場合、操作者以外のノードでは再生中の画面が突然停止箇所の画面に切り替わるなどといったことになる。

[早送り、巻き戻し] 早送り中（もしくは巻き戻し中）の情報については、それらの操作を行なっている操作者以外には必要ないと考えられる。つまり、操作者以外のノードは早送り（または巻き戻し）が終わった箇所さえかわればよいであろう。

[再生] 巻き戻し中に再生コマンドが発行される可能性もあり、すべてのノードにおいて操作者のノードで再生が開始された箇所から再生が開始されるためには巻き戻しコマンド以外に再生開始箇所を示す情報を送る必要がある。

[逆再生] 逆再生の場合も再生の場合と同様。

## 3. メディアの分類

会話的な操作を考える場合、一時停止、巻き

戻しなどの操作があるため、時間が逆行する可能性がある。現在表示されている（または聞こえている）情報が少し前の情報に依存する場合、時間が逆行した際に過去の箇所の情報を再現するには、現在から再現されるべき過去までの間の全ての履歴が必要であるかもしれない。

また、メディア間の同期ということ考えると、ある一定時間ごとにフレーム情報として情報が送られるメディアと、あるオブジェクトを右に3 cm/sの速度で移動させよなどといったコマンドで情報を伝えるメディアとを同期させるには（逆再生なども考え）何かしら工夫が必要になってくる可能性がある。

そこで、メディアを以下に示す3種類に分類する。

- (1) フレーム内圧縮のみのビデオ、音声などのような時間的に依存関係のないメディア。
- (2) ビデオのフレーム間圧縮などのように再生に履歴情報の必要なメディア。
- (3) CGなどであるオブジェクトの移動に関する情報を伝える場合のように、さらに受信ノードでの補間が必要なメディア。

#### 4. 同期手法

##### 4. 1. semantics

メディア間同期には以下に示すように幾つかのレベルを考えることができる。

- (1) 各メディアの再生がグローバルな時間軸上で正確に行なわれる。この場合、メディア間のタイミングも正確に合う。
- (2) メディア間のタイミングは正確に合わせるが、グローバルな時間軸上での再生位置のある程度のずれは許す。
- (3) メディア間のタイミング、各メディアのグローバルな時間軸での再生位置の両方共、ある程度のずれを許す。

(2)の方法を考える場合、時間軸方向にメディアを伸張、圧縮する技術が必要となる。ま

た、(2)、(3)の方法でどの程度のずれを許すかについては、人の主観的な判断に負うところが大きい。ずれを許すことで同期のsemanticsはより緩いものになる。

ここでは、(1)の最も厳しいメディア間同期について考える。

##### 4. 2. 同期の取り方

会話型操作を行ないながら同期を維持する方法について検討する。メディアの操作としては再生、一時停止、早送り、巻き戻しなどを考え、主にビデオ、音声などのような再生に履歴情報を必要としないメディアを想定して議論を進める。また、文献[5]と同じく、同期操作はアプリケーション層の操作とする。

会話的操作を行なう場合、グローバルな時間軸上で現在再生されている時点から過去（あるいは未来）の別の時点に跳ぶことがある。少しだけ巻き戻して一時停止する等といった場合、現在の時点から少し前後する時点までの情報をバッファ中に保っておけばバッファリングされている情報を用いることですぐにその時点に跳んでいける。

このように、過去の情報を保っておくことなど、前後情報保存用のバッファ(S-Buffer)に対する操作は複雑になるため、コーデック用のバッファ(C-Buffer)とS-Bufferを共通に用いることは会話型の操作を行なう場合難しいと考えられる。従って、会話型操作を行なわない環境で用いられるような音声／画像コーデックの受信バッファによりタイミングを保証する方式[6]は、この場合用いることができない。

また、会話型操作を行なわない環境ではバケット中に何かしらの時間情報を付加することでネットワークの揺らぎを吸収することのみを考えればよいが、会話型操作を伴う環境で同期を取る場合、バケット以外にバッファリングされている情報や操作のための情報にも何かしらの時間情報を付加する必要がある。

##### 4. 2. 1. バケットへの付加情報

会話型操作を伴う環境で同期を取るためにバ

ケットに付加する時間情報としては以下のようなものが考えられる。

- ＜絶対時間＞ 0から増え始めるカウンタ  
(つまり、メディア再生開始時点からの絶対時間)。
- ＜相対時間＞ 前後するフレーム間などの相対時間。
- ＜周期カウンタ＞ 一定周期で0に戻り、再び増え始めるカウンタ。
- ＜フレーム番号＞ フレーム番号。(フレーム間隔一定の場合)

まず、＜絶対時間＞の場合について検討する。この場合、絶対時間を指定できるため、巻き戻しの後に再生が再開される場合などにも対応できる。しかし、メディアの再生時間が長くなるにつれ必要なフィールドの大きさも増加する。例えば、25MHzクロックで、8クロックごと(0.32 $\mu$ sごと)にカウンタをインクリメントする場合、16bitの2進カウンタでは21msしか表せない。さらに長い時間を表すためにはカウンタがインクリメントされる間隔を広げるか、カウンタのbit数を増やすかのどちらかが必要となる。

＜相対時間＞については、＜絶対時間＞のような大きなフィールドを必要しないという利点がある。しかし、一つのケットが失われるとそれ以後のすべての時間が狂うなどの問題点がある。

＜周期カウンタ＞は＜絶対時間＞においてすべてのbitが1になった後で再び0から始めるというものである。この場合、＜絶対時間＞と同じフィールド長で無限時間を表すことができ、また、ケットが失われても時間が狂うことはない。ただし、巻き戻しの後に再生が再開される場合などには対応しておらず、その場合、一度絶対時間を送る必要がある。また、無声区間などでケットが21ms以上送られないような場合にも受信側での累積時間(絶対時間)が増えるように空ケットを(少なくとも21msに一度)送る必要がある。

＜フレーム番号＞はフレームごとに(1ずつ増える)番号を付けるため必要なフィールド長は比較的小さくてすむ。また、サンプリング周波数が音声などのように比較的大きい場合、

フィールド長はかなり大きくなるが、この場合は＜周期カウンタ＞と同様な方法を適用できる。ただし、この方法はフレーム間隔が一定のメディアにしか用いることができない。

ケットに付加する情報の場合、フィールド長はあまり大きくない方がいいであろう。また、相対時間を送る方法はケット紛失で決定的なダメージを被ってしまう。以上の検討から、等間隔で情報が発生するメディアに対しては＜フレーム番号＞、情報発生間隔が変動するメディアでは＜周期カウンタ＞がよいと考えられる。

また、再生に履歴情報を必要とするようなメディアでは、上記と同じ方法で対応できる。受信ノードでの補間が必要なメディアでは絶対時間を用いて動作を記述することが必要であり、＜絶対時間＞のような絶対時間と移動速度などを組み合わせて用いるのがよいであろう。

#### 4. 3. 2. 操作のための情報への付加情報

つぎに、操作のための情報に付加する時間情報について検討する。操作のためのプロトコルは5節で示すが、必要となる時間情報は時間軸上での停止位置、再生開始位置などである。

この場合に用いる時間情報は時間軸上で絶対的な値である必要がある。よって、＜相対時間＞＜周期カウンタ＞を用いることはできない。そこで、等間隔で情報が発生するメディアに対しては＜フレーム番号＞、情報発生間隔が変動するメディアでは＜絶対時間＞を用いることが考えられる。しかし、各メディアごとに別々の時間情報を用いるよりも一つの時間情報で停止位置を表したほうがケットが短くて済むこと、及び、＜フレーム番号＞では時間間隔が違うメディア毎に別々の数を用いる必要があることなども考慮し、再生に履歴情報の必要なメディア、受信ノードでの補間が必要なメディアなども含めてすべてのメディアに対し適用できる＜絶対時間＞の方法が適当と考えられる。

#### 4. 3. 3. バッファリングされている情報への付加情報

つぎに、バッファリングされている情報に付

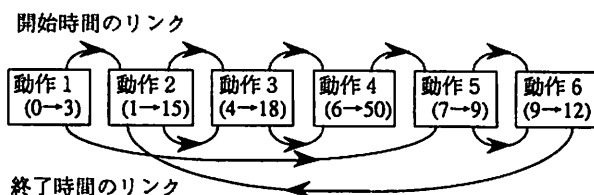


図2：動作バッファの管理

加する時間情報について検討する。この場合も、用いられる時間情報は時間軸上で絶対的な値である必要がある。よって、＜相対時間＞＜周期カウンタ＞を用いることはできない。そこで、等間隔で情報が発生するメディアに対しては＜フレーム番号＞、情報発生間隔が変動するメディアでは＜絶対時間＞を用いるのが適当であろう。

また、再生に履歴情報を必要とするメディアでは、上記と同じ方法で対応できる。受信ノードでの補間が必要なメディアでは絶対時間を用いて記述された動作をバッファリングする必要がある、＜絶対時間＞のような絶対時間と移動速度などの情報を組み合わせて用いるのがよいであろう。また、この場合、巻き戻しなどの操作で時間が跳ぶ場合に、現在の時点から過去のある時点までに行なわれた動作がどれであるのか識別する必要がある。一つの方法として図2に示すような動作の開始時間と終了時間をその値でソートして繋ぎ合わせたリンクを利用することが考えられる。

また、再生に履歴情報を必要とするメディアでは、巻き戻しなどの操作で時間が大きく跳んだ時に必要な情報量はかなり大きくなり、必要な動作もかなりの量になるため、ある程度の時間おきに全情報を持つことで依存関係を切ることも必要であろう。

## 5. メディア操作のためのプロトコル

一時停止、巻き戻しなど幾つかの操作について、そのプロトコルを検討する。

### 5. 1. 一時停止

一時停止の場合のプロトコルを以下に示す。

(案1) 情報発生源のみに一時停止コマンドを送る。他のノードは情報発生源から(履歴情報の必要のないメディアの場合は停止位置での(フレーム)情報と共に)一時停止の通知を受ける。バケット中に含まれる情報としては一時停止コマンドが発行された(グローバルな時間軸上での)位置が必要。

(案2) 全てのノードに一時停止コマンドをブロードキャストする。履歴情報を必要としないメディアの場合は停止位置での(フレーム)情報が情報発生源から各ノードにブロードキャストされる。バケット中に含まれる情報は(案1)と同じ。

(案1)では(案2)より少ないバケットで処理が完了する。しかし、各ノードは情報発生源から一時停止を通知されるまで再生を続けてしまうために、各ノードでバッファリングされている情報が先に進み過ぎてしまい、最悪の場合、バッファ中の情報だけでは停止位置を再生できなくなる。(案2)では(案1)よりも早く各ノードが一時停止を通知されるために各ノードでバッファ中に停止位置付近の情報を維持できる。従って、(案1)よりもバッファ中の情報だけでは停止位置を再生できなくなることが起こりにくい。

以上の検討より、(案2)の方法を用いることが適当であると考えられる。

### 5. 2. 巻き戻し

巻き戻しのプロトコルを以下に示す。

(案1) 情報発生源のみに巻き戻しコマンドを一度送る。

(案2) 全てのノードに巻き戻しコマンドを一度送る。

(案3) 全てのノードに巻き戻しコマンドを定期的に送る。

(案1) は一時停止の場合と同様に、情報発生源以外のノードが巻き戻し中であることを即座に知ることができないために良い案ではない。また、情報源以外のノードは一度巻き戻し中であることがわかれば、後は未来の情報を捨て、情報源から送られてくる情報を受け入れるしかないの、(案3) のように二度以上巻き戻しを知らせても意味がない。よって、(案2) が適当であると考えられる。

また、巻き戻しの場合、操作者以外が巻き戻し中の画面を見る必要はないと考えられる。従って、再生に履歴情報を必要としないメディアに関しては巻き戻し中には操作者以外のノードに情報源からパケットを送る必要はない。ただし、再生に履歴情報が必要なメディアでは、巻き戻しのためにも履歴情報が必要なため、パケットを送り続ける必要がある。

### 5. 3. 早送り

早送りのプロトコルに関しては、巻き戻しと同様、(案2) のプロトコルを用いればよい。

また、巻き戻し、早送りのどちらの場合にも、操作者のノードの画面では通常よりも早く(しかも巻き戻しでは反対方向に)画面が動いていく必要がある。但し、この場合、きれいな画面である必要はないため、再生に履歴情報の必要ないメディアではフレームを間引くなどの方法で対処すれば十分であろう。再生に履歴情報の必要なメディアでは、MPEGのように周期的に完全なフレームが入っていればそれだけ送れば良いが、履歴が絶ち切れなような場合は表示速度を上げるのは難しいであろう。また、受信ノードでの補間が必要なメディアでは、(右に3cm/sで動けなどといった)動作指定コマンドの速度を数倍にすれば良い。逆方向への動作についてはコマンドを逆方向に解釈する必要がある。

### 5. 4. 再生

再生コマンドに関しても各ノードにブロードキャストするのが適当と考えられる。パケット中に含まれる情報としては、一時停止していた状態からだけではなく早送り中にいきなり再生ボタンが押される可能性も考えると、再生コマンドが発行された(グローバルな時間軸上で)位置が必要となる。

また、一度再生が始まると、パケット遅延の変動がなければ図1でfutureの大きさは再生の間一定に保たれ、変わることも、また意図的に変えることもできない。そのために、再生開始時点でその値を決めておくことが重要になる。そこで、各情報発生源でパケットを送り始めるタイミングと、各ノードで再生を実際にスタートさせるタイミングについて考える。

まず、各情報発生源でパケットを送り始めるタイミングについて考える。パケット送出開始のタイミングの決め方を以下に2つ示す。

(方法1) 各情報発生源が再生命令を受け取った時点でパケットを送り始める

(方法2) 各情報発生源が再生命令を受け取り、そこに書かれている再生開始時間に各情報発生源での理想的なオフセット時間を足した時間からパケットを送り始める。

(方法1) は単純であるが、各ノードであまりに各メディアの到着がばらばらだと早く到着するメディアのfutureの部分にメモリを大きく割く必要があり、到着時間の揺らぎを吸収するた

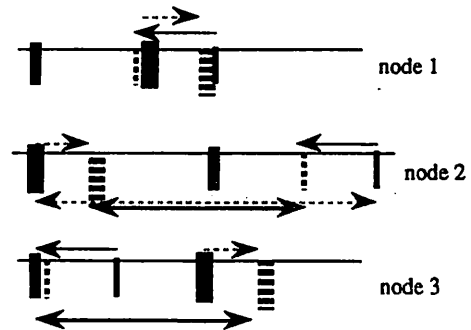


図3：オフセットの決め方

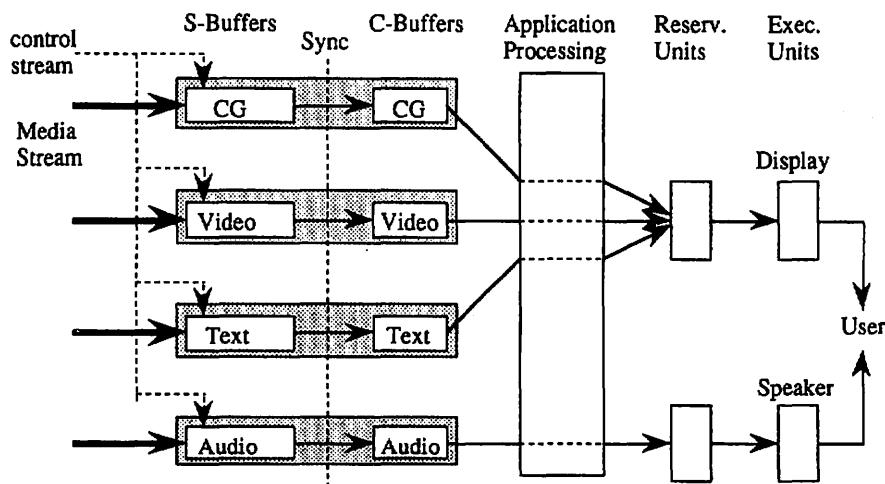


図4：端末構成

めにメモリをあまり使えなくなってしまう。

(方法2)では、全てのノードでの同期を考慮した最適な各情報発生源の再生開始時間があることを仮定している。再生命令を受け取った時点で既にそこに書かれている再生開始時間が過ぎていることがなく、かつ、再生開始までに時間がかかりすぎないように再生開始時間を設定することが難しいこと、及び、理想的なオフセット時間を予め決定しておく必要があるものの、各ノードでのタイミングは(方法1)に比べれば比較的うまくとれるであろう。理想的なオフセット時間の決め方としては、まず、各情報源から同時に再生が開始された場合に各ノードに到着する各メディアの到着時間のずれを観測する。各ノードでのメモリ量が等しい場合を考えると、まず、ある2つのメディアの到着時間のずれが最も大きいノードを考え、その2つのメディアを近づける方向に動かす(つまり、オフセット時間を入れる。)。この際に、このノードでの2つのメディアのずれと他のノードでの2つのメディアずれの最大値をみていて、他のノードでのずれと等しくなったところで止める(図3)。さらに2つのメディアの到着時間のずれが大きいノードに対しこれを繰り返すことで理想的なオフセット時間を決める。

次に、各ノードで再生をスタートさせるタイ

ミングについて考える。各ノードでは再生コマンドを受け取った時点で再生可能状態に入る。さらに各情報発生源からの再生方向へのパケットが到着し始め、futureの部分の情報が適度な量になった時点で再生を開始させるのがよいであろう。再生開始点の決め方としては、最初に到着したメディアのpastの(時間)量と最後に到着したfutureの(時間)量がほぼ等しくなるような時点で再生を開始する、といった方法が考えられる。

## 6. Packet Format

次に、Packet Formatについて簡単に触れる。

- ・ Interchange Message := Envelope + Content;
  - ・ Envelope
    - ・ Media Id
    - ・ Time information
    - ・ アドレス情報(送信側, 受信側)
    - ・ (transform情報)

Time informationは5. 3. 1. 節で検討した付加情報を用いる。Media Idは各メディアを識別するために必要であり、大規模なネットワーク環境で用いるような場合、一つのノードが複数

の（同期）グループに属している場合が考えられるため、Group Idとグループ内でのMedia Idの組を用いる必要がある。

また、例えば英語の音声がある通訳ノードで日本語の音声に変換されて配送されるなどといった、メディア変換を考える場合、通訳ノードのアドレスとtransform情報が必要になる。

## 7. ノード構成

各ノードのシステム構成を図4に示す。

## 8. おわりに

各メディアが別々の通信路（またはコネクション）を通して伝送されるようなマルチメディア通信システムにおいて、会話的な操作を行なう場合にパケットなどに付加すべき時間情報、及び、会話的操作を行なうためのプロトコルを示した。

今後はこのようなシステムにおけるバッファ量と応答性の関係などについてさらに検討を進めると同時に、システムの試作についても行なっていく予定である。

## 参考文献

- [1]秋山, 他, " 広帯域ISDN特集", 電子情報通信学会誌 1991-11.
- [2]安田浩著, " マルチメディア符号化の国際標準", 丸善, 1992年6月.
- [3]Thomas D.C.Little, "Multimedia Synchronization Protocols for Broadband Integrated Services", IEEE Journal on selected areas in communications. vol.9. no.9., December 1991.
- [4]Ralf Steinmetz, "Synchronization Properties in Multimedia Systems", IEEE Journal on selected areas in communications. vol.8. no.3., April 1990.
- [5]Klara, Jonathan, "An Integrated Multimedia Architecture for High-Speed Networks", Multimedia '92.
- [6]中島, 三浦, 矢部, " ATM網を用いたTV会議システムにおけるタイミングについての一検討", 信学技報 OS91-11.