

並列分散型連続メディア処理アーキテクチャ HEARTS

小倉 毅

小野 諭

高橋 直久

{ogura, ono, naohisa}@slab.ntt.jp

NTT ソフトウェア研究所

要旨

分散環境で連続メディアを高品質に処理するには高精度の周波数同期が不可欠であるが、周波数情報をデータと共に送受する従来の方法では簡単な網構成の場合にしか適用できない。本稿では、高速デジタル通信網のクロックを用いた、網構成に依存しない周波数同期技術により非同期な分散環境で高品質の連続メディア処理を可能にする HEARTS アーキテクチャを紹介し、複数ストリームの同時性保証通信を実現する方法を述べる。

1 はじめに

近年、ビデオ画像、音声、リモートセンシング波形などの連続メディアを計算機ネットワーク上で伝送/処理するための技術開発が盛んに行なわれている。このような連続メディアはそれ自身時間軸に関する情報をもった同期ストリームであり、高品質な連続メディア処理をおこなうためには、これらの時間軸を正確に扱う必要がある。

メディアの時間軸を正確に扱うための方法として、送受信地点間で終始同期的な伝送/処理をおこない、メディアの時間軸をそのまま実際の時間経過として保存する方法が考えられる。この方法は、ジッタの発生に敏感な情報を扱うのに適しており、現在のデジタル電話などでも用いられている。

しかし、このような方法はジッタを非常に小さくすることができる反面、サービス帯域が固定されてしまう。これは、普段使わない帯域も常時確保しておくことを意味するため、伝送路の利用効率が低くなるという問題がある。また、情報源の速度変化に対応できないという問題や、ジッタを許容しないために可変ビットレート処理、蓄積処理、汎用計算機処理との共存ができないという問題がある。

非同期通信では、上記のような同期処理がかかえる問題を解決することができるが、伝送路における

ジッタを許容するため、メディアの時間軸を受信側で再現する必要がある。このとき、送受信地点間での各種デジタル処理装置のクロック周波数同期の精度がメディア処理の品質に大きく影響する。非同期通信網におけるクロック周波数同期手法として、従来から MPEG や queuing feedback といった方法が存在する。しかし、これらの方法は、クロック周波数情報をデータと一緒に送受するため、適用できるネットワークポロジが簡単なものに限定されてしまうという問題がある。

以上のような問題をふまえ、筆者らは非同期処理をベースとした分散環境の上で同期ストリームの高品質な処理を可能にするための HEARTS (Heterogeneous Asynchronous Real-Time Stream computing) アーキテクチャを提案している [1]。HEARTS アーキテクチャは、ネットワークポロジに依存しない広域周波数同期技術を用い、システムをこの周波数に同期して動作する同期処理部と、ジッタを許容する非同期処理部とに分離する。そして、両者の境界でタイムスタンプによりメディアの時間軸の保存、回復をおこなう。非同期処理の利点を生かしながら、必要ところで高精度の同期処理を可能にするアーキテクチャである。

本稿では、提案する HEARTS アーキテクチャを紹介し、HEARTS アーキテクチャの構成要素を用いて連続メディアの時間軸を再現する手法を述べる。具体的には、遠隔地で発生する複数の同期ストリームを扱う際に、各地点でサンプリングしたデータ相

Parallel and distributed continuous media processing architecture "HEARTS".
Tsuyoshi OGURA, Satoshi ONO, Naohisa TAKAHASHI.
NTT Software Laboratories.

互の時間軸上の位置関係を受信側で正確に再現する同時性保証機構について述べる。本手法は、各ストリームの等時性の保証も実現できること、遅延が有界なときにフィードバック系のフロー制御なしに最小遅延、最小バッファ容量でデータ落ちのない同時性保証通信を可能にすることを特徴とする。

本論文の構成は以下の通りである。章 2 で従来の周波数同期技術とその問題点を述べる。章 3 で HEARTS アーキテクチャの基本概念と構成要素を説明し、章 4 で現在考案している基本要素の実現方法を述べる。章 5 では、HEARTS の構成要素を用いて同時性保証機構を実現する方法を説明する。最後に、章 6 でまとめをおこなう。

2 従来の周波数同期技術

非同期通信網ではジッタを許容するため、一度失われたメディアの時間軸をあとから回復する必要がある。このとき、送信地点と受信地点のクロック周波数同期の精度がメディア処理の品質に大きく影響する。

非同期通信をベースとしたシステムでは、同期通信の場合のようにデータの到着間隔から送信地点のクロックを回復することができない。そのため、各種処理装置はそれぞれ独立したクロックで動作する。このとき、各処理装置のクロック同期の精度が十分でないとき、誤差が蓄積し、メディアストリームを長時間再生すると時間的なずれが生じたり、スリップが発生する。例えば、標準的に用いられている水晶発振器の精度は 10^{-5} 程度であり、連続メディアをフルデジタル処理するのに十分な精度とはいえない。

このような問題に対する解決策として、以下の 2 つの方法がある。

1. 高精度クロックによる完全独立同期
2. 非同期処理の中へ同期技術を埋め込む方法

1 は、各送/受信地点に Cs などの超高精度の発振器を置く方法である [2]。各地点がお互いに完全に独立で構成が簡単になる。しかし、各地点に高精度のクロックを置くため、コストが高くなる。

2 に属する方法としては、queuing feedback と呼ばれる方法 [3] や、MPEG で用いられる周波数同期の方法 [4] がある。

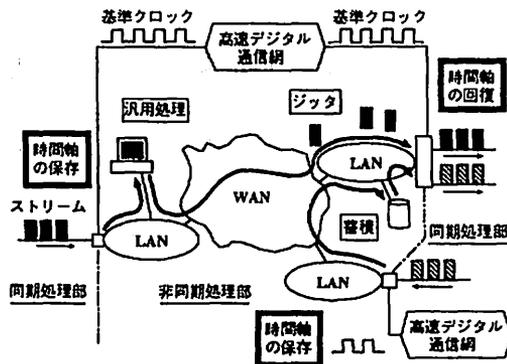


図 1: HEARTS アーキテクチャの概念図

queuing feedback は、受信側が自身のキューにたまったメッセージ長を送信側に送り、送信側はその値に応じて自身のクロック周波数を制御する方法である。この方法は、送信側のクロック周波数を受信側の要求に応じて調整するため、放送などの point-to-multipoint 型の通信では利用できないという問題がある。

MPEG では、符号器側で SCR (System Clock Reference), PCR (Program Clock Reference) といった周波数情報をストリーム内に埋め込み、復号器側はそれらの値に基づいてクロック周波数を回復する。この方法は、復号器側のクロック周波数を符号器側に同期させるため、multipoint-to-point 型の通信や multipoint-to-multipoint 型の通信では利用できないという問題がある。

3 HEARTS アーキテクチャ

前章のような問題をふまえ、筆者らは非同期処理をベースとした並列分散環境で高精度な同期ストリーム処理を可能にする HEARTS アーキテクチャを提案している。本章では、HEARTS アーキテクチャの基本概念、および、現在までに考察が終了している処理モデルの概要を述べる。

3.1 基本概念

筆者らの提案する HEARTS は、音声、映像、数値などの性質の異なる (Heterogeneous), 処理や伝送が相互に非同期 (Asynchronous) に行なわれる、実

時間制約 (Real-Time) のある, 連続的 (Stream) なデータの流れを広域分散環境下で並列に処理するためのアーキテクチャである。

図 1 に概念図を示す。HEARTS では, システムを高精度な周波数源に同期して動作する同期処理部と, ジッタを許容する非同期処理部とに明確に分離する。非同期処理部の前段で連続メディアストリームの処理単位 (以下, パケットと呼ぶ) 毎にタイムスタンプを付与しながら時間軸情報を保存する。そして, 非同期処理を行なったあと, 必要なところでパケットをタイムスタンプに従って出力し, 時間軸を回復する。

メディア処理の品質を高めるためには, 上記タイムスタンプ処理を高精度におこなう必要がある。このためには, 広域に分散する同期処理部の処理装置の動作クロック周波数を高い精度で同期させる必要がある。HEARTS では, 章 2 で述べた従来手法のように非同期処理の中でクロックに関するデータを送受するのではなく, 後述する高精度周波数源を用いて常時同期処理部に高精度周波数を供給する。これにより, ネットワークのトポロジに依存しない形で高精度な広域周波数同期を実現する。

メディアストリームの時間軸情報を実際の時間経過でなくデータとして処理するので, 汎用の計算機処理, 可変ビットレート処理, 蓄積処理などの処理との共存も可能であり, 非同期処理の利点を生かしながら高精度な周波数同期を実現することが可能である。

3.2 広域周波数同期

広域分散環境における高精度周波数源として, 筆者らのグループでは, 高速デジタル網を利用した周波数同期方法 [5] を提案している。これは, ISDN などの高速デジタル網のフレーム同期ピットのタイミングから生成したクロックをシステム内の計算機ノードや各種デジタル処理装置で共有することにより周波数同期を図るというものである。この方法では, point-to-point, point-to-multipoint, multipoint-to-multipoint などのトポロジに依存しない高精度なクロック周波数同期方法が可能となる。HEARTS アーキテクチャではこの広域周波数同期技術を用いる。

現在, INS 64 のフレーム同期クロック (IBFC:

ISDN BRI Framing Clock) を用いた周波数/時刻同期システムが稼働しており, 周波数/時刻同期について精度評価が終了している [6]。その結果, 周波数同期については 1.6×10^{-11} と非常に高精度の同期が可能であることがわかっている。

3.3 処理モデルの概要

一つの送信地点から一つの受信地点へメディアストリームを伝送する場合について, 処理モデルの概要を図 2 に示す。以下, 図中の構成要素について説明する (図では, 同期処理部に属する伝送路を太実線で, 非同期処理部に属する伝送路を太破線で示している。以降, 同様の記法を用いる)。

基準クロック供給装置

ISDN などの高速デジタル通信網の物理層から, フレーム同期クロックを抽出する。そして, このフレーム同期クロックを所望の周波数になるように分周または通倍した基準クロックを生成し, 同期処理部の他の構成要素に供給する。図 2 に示すように, HEARTS では A/D, D/A 変換, パケット生成など, メディアの時間軸を与えるすべての処理装置に対してこの基準クロックを供給することによって周波数同期を図る点が特徴である。

非同期化部

同期処理部から非同期処理部へ移る境界におく。基準クロックをカウントとする FSC (Frequency Synchronized Clock) を内部に持つ。サンプリング開始時刻 X , ストリームの継続時間 ΔX の二つのパラメータにもとづいて, 自身の FSC の値が X から ΔX の間, 同期処理部から到着するパケットの入力時の FSC の値をタイムスタンプとして付与し, 非同期処理部へ送出する。

同期化部

非同期処理部から同期処理部へ移る境界におく。基準クロックをカウントする FSC を内部に持ち, パケットのタイムスタンプ値 X , ストリームの継続時間 ΔX , 出力開始時刻 Y の三つのパラメータにもとづいて, 受信したパケットのうちタイムスタンプ値 t が $X \leq t \leq X + \Delta X$ なるものを FSC 値が

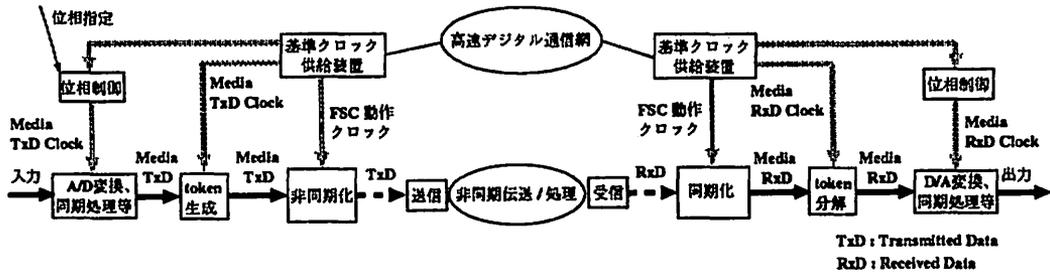


図 2: 処理モデルの概要

$(Y + t - X)$ の時点で同期処理部に送出する。すなわち、各パケットのタイムスタンプ値とそのデータを出力する時点での自身の FSC の値とのオフセットが、パラメータ X と Y のオフセットと常に等しくなる形で出力する。

等時性の再現

図 3 のように、FSC の間隔 δ だけ間をおいてサンプリングされた 2 つのパケットがあるとする。同期化部の動作パラメータの設定が図のような場合、2 つのパケットはそれぞれ同期化部の FSC が $Y + t - X$ 、 $Y + (t + \delta) - X$ の時に出力される。つまり、同期化部でも間隔 δ をもって出力される。ここで、両者の FSC のクロック周波数は基準クロックにより高い精度で同期しているので、両者の FSC の間隔 δ は実時間としても高い精度で一致している。したがって、ストリーム内のデータ相互の時間間隔を非同期処理のあとでも正確に再現することができる。

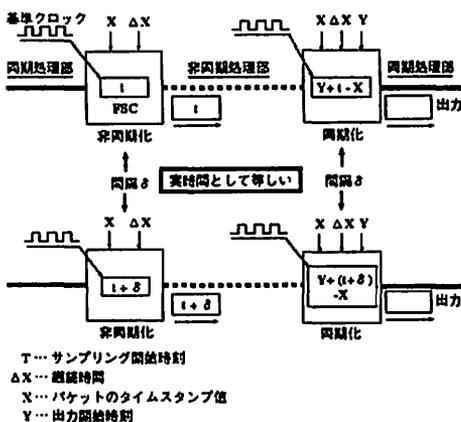


図 3: 非同期化/同期化と等時性の再現

4 基本要素の実現法

本節では、HEARTS の基本要素のうち、非同期化部と同期化部の実現方法を示す。

4.1 非同期化部の実現法

非同期化部の実現法を第 4 図に示す。タイミング生成回路は、パケットを入力した時点でラッチ回路に FSC の値をラッチするよう指令を出す。ラッチ回路は、この指令によって、FSC の値をラッチし、その値を FIFO である FSC 値保存用メモリに格納する。入力したパケットは、FIFO である入力バッファメモリに格納する。出力回路は、入力バッファメモリと FSC 値保存用メモリの先頭からそれぞれパケットとタイムスタンプ値を取り出し、パケットにタイムスタンプを付与して出力していく。動作開始/終了回路は、FSC の値がサンプリング開始時刻 X と等しい値になった時点で出力開始の指令を出し、FSC の値が $X + \Delta X$ になった時点で出力終了の指令を出す。

4.2 同期化部の実現法

同期化部の実現法を第 5 図に示す。バッファリング判定回路により、タイムスタンプ値が X から $X + \Delta X$ の間のパケットをバッファリングする。出力が開始されるまでの間、入力したパケットはタイムスタンプの古い順にソートされてバッファメモリ内に格納されていく (非同期処理の影響によりパケットの送信と受信の順序が異なることを仮定している)。FSC の値が Y になると、出力開始回路が出力判定回路に動作の開始を指令する。出力判定回路は、バッファメモリ内の先頭パケットのタイムスタンプ

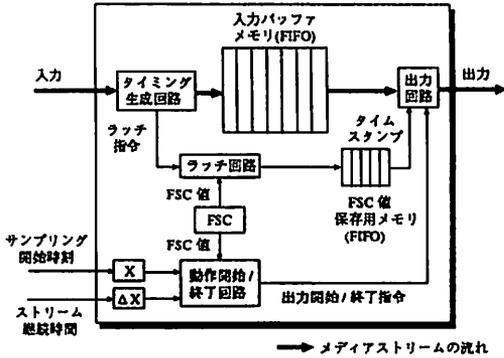


図 4: 非同期化部の実現法

値と、 $X - Y$ に現在の FSC 値を加えた値 (これを Z とする) とを比較し、両者が一致したときこのパケットからタイムスタンプを削除したパケットを出力し、バッファメモリから削除する。以下、FSC の値が $Y + \Delta X$ になるまでこの動作を繰り返すことにより、3.3 節で述べた定義にもとづいた出力動作をおこなうことができる。

5 同時性保証通信の実現

HEARTS アーキテクチャによって連続メディアの時間軸を再現する例として、遅延が有界である複数の送受信地点の組から成るシステムにおいて送信側でのデータ相互の時間軸上の位置関係を受信側で正確に再現しながら出力する同時性保証通信の実現法を述べる。本手法は、各ストリームの等時性保証も実現しながら、フィードバック系のフロー制御なしに最小遅延、最小バッファ容量でデータ落ちのない同時性保証通信を可能にする。

5.1 設計方針

1 ~ 6 で定められるシステムにおいて、無限長でフレーム境界のない速度 V' の N 本のストリームに対して、(1) ~ (6) の特徴をもつ同時性保証通信機構を実現することを考える。

1. 送受信地点の組の数 N
2. 各組の最大通信遅延時間 $D_{max i} (1 \leq i \leq N)$
3. 各組の最小通信遅延時間 $D_{min i} (1 \leq i \leq N)$

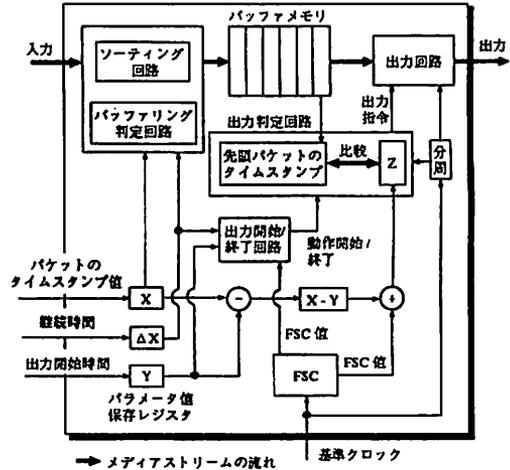


図 5: 同期化部の実現法

4. 各組の通信路の帯域 $V (\geq V')$
5. 各受信地点のバッファ容量 $B_i (1 \leq i \leq N)$
6. 制御パラメータを全ての送受信地点に通知するのにかかる最大時間 U

- (1) ユーザが指定した送信側のデータのサンプリング開始時刻 t_1 、受信側の出力開始時刻 t_2 に対して、以下の (a) ~ (c) の異常の有無を判定する。さらに、異常がないとき、同時刻にサンプリングしたデータを同時刻に出力する。

- (a) t_1 が小さすぎていずれかの送信地点でサンプリングミスが生じる可能性がある
- (b) $t_2 - t_1$ が小さすぎていずれかの受信地点でデータの欠落が生じる可能性がある
- (c) $t_2 - t_1$ が大きすぎていずれかの受信地点でデータの欠落が生じる可能性がある

- (2) 受信側から送信側にフィードバックをかけるようなフロー制御をおこなう必要がない
- (3) 各ストリームの等時性が回復できる
- (4) ユーザが t_1 の指定を省略した場合、上記 (1) の (a) の異常が発生しないことを保証しつつ t_1 をできるだけ早い時刻に設定できる
- (5) ユーザが t_2 の指定を省略した場合、上記 (1) の (b) の異常が発生しないことを保証しつつ t_2 をできるだけ早い時刻に設定できる

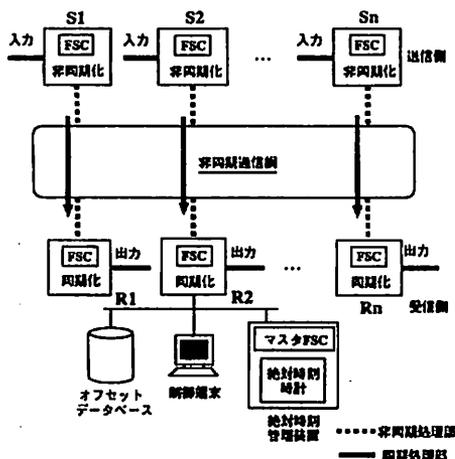


図 6: 同時性保証システムの全体構成

(6) 広域分散制御を可能にするため、各送受信地点で時刻管理がローカルにおこなえる

5.2 システムの概要

同時性保証通信システムの構成を図 6 に示す。非同期化部/同期化部 (以下、 S_i, R_i) を各送受信地点に設置する。前節 (6) にもとづき、 S_i, R_i の各 FSC はローカルな値をもつ。絶対時刻管理装置とオフセットデータベースは、ユーザが指定する絶対時刻 t_1, t_2 と S_i, R_i の各 FSC の値を対応づけるために用いる。次節でこれらについて説明する。

5.3 指定時刻と FSC の対応

ユーザは制御端末から送信側でのサンプリング開始時刻 t_1 、受信側での出力開始時刻 t_2 を指定する。 t_1, t_2 が指定されると、 S_i のサンプリング開始時刻、および、 R_i のパケットのタイムスタンプ値に t_1 に相当する FSC の値 (それぞれ、 $C_{S_i}(t_1), C_{R_i}(t_1)$ と表記) を、また、 R_i の出力開始時刻に t_2 に相当する FSC の値 ($C_{R_i}(t_2)$) をセットする。これらの FSC の設定は以下のようにしておく。

制御端末は、 t_1, t_2 におけるマスター FSC の値 $C_{mas}(t_1), C_{mas}(t_2)$ の計算を絶対時刻管理装置に依頼する。絶対時刻管理装置は、依頼があった時刻 t における自身の絶対時刻時計の値 $A(t)$ 、および、

マスター FSC の値 $C_{mas}(t)$ をラッチする。そして、これらの値と基準クロックの動作周波数をもとに、 $C_{mas}(t_1), C_{mas}(t_2)$ を求め制御端末に渡す。

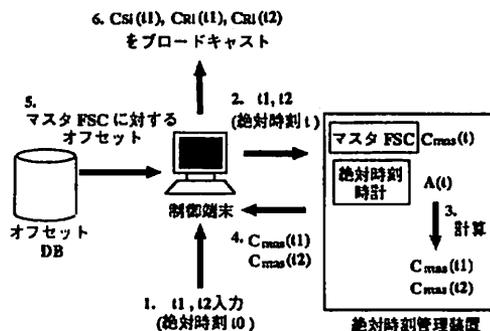


図 7: 時刻同期

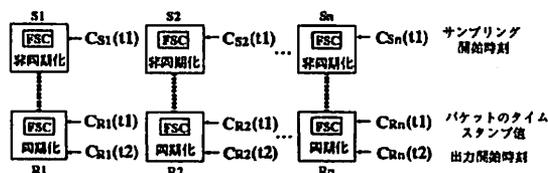


図 8: 非同期/同期化部のパラメータ設定

各 FSC は基準クロックによって周波数同期がかかっているため、それらの値のオフセットは常時一定である。そこで、あらかじめ各 FSC のマスター FSC に対するオフセットをデータベース化しておく (オフセットの測定方法については [6] を参照されたい)。端末は、このオフセットデータベースにアクセスし、 $C_{mas}(t_1), C_{mas}(t_2)$ の値と、各 FSC のマスター FSC とのオフセットから、 $C_{S_i}(t_1), C_{R_i}(t_1), C_{R_i}(t_2)$ を求め、ブロードキャストする。

以上述べた手順を図 7 に、FSC 値の設定結果を図 8 に示す。この方法により、5.1 節の (6) を満足しながら S_i, R_i の各パラメータを同一時刻の FSC の値に設定することができる。

5.4 同時性保証の原理

前節の手順により、各 S_i のサンプリング開始時刻、および、各 R_i のパケットのタイムスタンプ値、

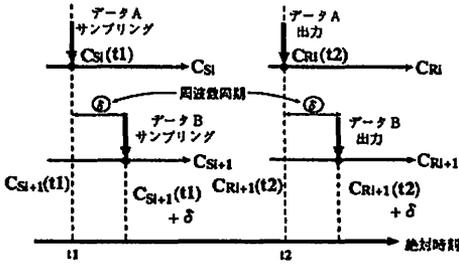


図 9: 同時性保証の原理

出力開始時間をそれぞれ同一時刻に同期した FSC の値に設定することができる。このとき、3.3 で述べた非同期化部/同期化部の動作定義により、図 9 のように、非同期化部 S_i, S_{i+1} において FSC の基準クロックサイクル δ の間隔をおいてサンプリングされた 2 つのデータは、受信側の R_i, R_{i+1} においても δ の間隔をおいて出力される。送信側、および、受信側における δ は、FSC の周波数同期により実時間としても等しい時間間隔なので、2 つのデータのサンプリング時刻の差が出力時にもそのまま再現される。したがって、同時性の保証が可能となる。また、1 本のストリームについてみたとき、3.3 節で述べた非同期化部/同期化部の周波数同期により等時性の保証も実現される。

各 FSC の動作周波数が高い精度で同期しているので、5.1 節 (1) の (a)~(c) の異常が発生しないことを保証できれば、以後受信側からフィードバックをかけるようなフロー制御をおこなわなくてもデータが欠落することなく同時刻にサンプリングしたデータを必ず同時刻に出力することができる。すなわち、5.1 節の (1), (2), (3) を満足することができる。

次節では、まず t_1, t_2 の値にかかわらず網の遅延によって定まる S_i, R_i に必要な最小のバッファ容量を述べる。そして、5.1 節 (1) の (a)~(c) の判定、および、(4), (5) を実現するための t_1, t_2 の条件を述べる。

5.5 最小バッファ容量

網の伝送遅延によって、 S_i, R_i でバッファのあふれによるデータの欠落を起こさないために必要なバッファ容量の最小値が以下のように定まる。

S_i から送り出されるパケットは最大 $D_{max i}$ の間

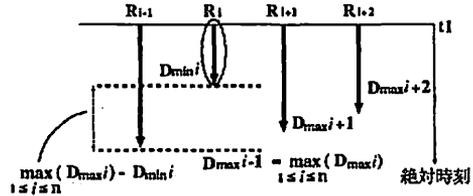


図 10: パケット到着時間のばらつき

S_i 中に滞留する可能性がある。したがって、この間に S_i 上でバッファのあふれを起こさないためには、以下の容量のバッファが S_i に必要となる。

$$D_{max i} \times V$$

次に、各 R_i でバッファのあふれを起こさないために必要なバッファ容量の最小値を述べる。各 R_i は、ストリームの先頭パケットが到着した時点から全ての R_i に先頭パケットが届くことが保証できる時刻までに最長

$$\max_{1 \leq i \leq N} (D_{max i}) - D_{min i}$$

だけ待つ可能性がある (図 10)。この間に R_i 上でバッファあふれを起こさないためには、上記の値に V を掛けた

$$V \times \left\{ \max_{1 \leq i \leq N} (D_{max i}) - D_{min i} \right\}$$

のバッファが R_i に必要となる。

帯域 V の通信路を流れるストリームの速度 $V' (\leq V)$ がコネクション確立時に申告されるようなプロトコルを用いる場合には、上述した容量のバッファを S_i, R_i に実装し、コネクション確立時に上述の V を V' に置き換えた容量の領域をバッファ領域として動的に確保すればよい。

5.6 指定時刻の設定方法

5.1 節 (1) の (a) ~ (c) の異常が生じないことを保証するため、および、(4), (5) の設定をおこなうために t_1, t_2 が満足すべき条件式を示す。

(1)-(a)

ユーザが時刻 t_0 に t_1, t_2 を指定してから、全ての送受信地点への制御パラメータの伝達をサンプリング開始時刻 t_1 までに終了すればよいので、以下の式を満たせばよい。

$$t_0 + U \leq t_1$$

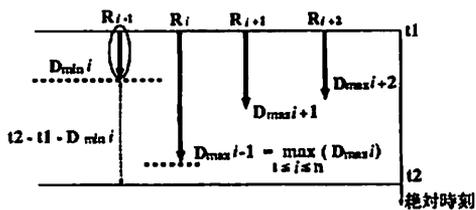


図 11: サンプリング/出力時刻と遅延時間の関係

(1)-(b)

時刻 t_1 にサンプリングを開始してから時刻 t_2 までに全ての R_i ヘストリムの先頭パケットが到着していることを保証すればよい。それには、 t_1 と t_2 の時間間隔が各送受信地点間の最も長い最大遅延時間以上あればよいので、以下の式を満たせばよい。

$$\max_{1 \leq i \leq N} (D_{max i}) \leq t_2 - t_1$$

(1)-(c)

t_1, t_2 の間に R_i に届くデータ量は $V' \times (t_2 - t_1 - D_{min i})$ 以下なので (図 11), 以下の式を満たせばよい。

$$\forall i \quad t_2 - t_1 - D_{min i} \leq B_i / V'$$

(4)

(a) についての式の不等号を等号におきかえたときの t_1 を用いればよい。

(5)

(b) についての式の不等号を等号におきかえたときの t_2 を用いればよい。

ユーザから t_1, t_2 の指定があった時に、それらが以上の判定式を満足することを判定することにより、5.1 節の (1) - (a) ~ (c), (4), (5) を実現することができる。上記の判定に必要な $U, D_{max i}, D_{min i}, B_i$ ($1 \leq i \leq N$) もあらかじめデータベース化しておけばよい。

6 おわりに

本稿では、非同期な分散環境における連続メディア処理のためのアーキテクチャ HEARTS を紹介した。そして、連続メディアの時間軸を正確に再現する例として、複数地点で同時刻にサンプリングしたデータを受信側で正確に同時刻に出力する同時性保

証通信の実現方法を述べた。本手法によって、ネットワークの遅延が有界であるという仮定のもとで、最小遅延、最小バッファ容量で、フィードバック系のフロー制御なしに、データ落ちのない同時性保証通信を実現することが可能となる。

HEARTS アーキテクチャは、連続メディアの時間軸を実際の時間経過ではなくタイムスタンプデータとして扱うこと、高速デジタル網のクロックを用いた周波数同期技術によりネットワークポロジに依存しない高精度な周波数同期を実現すること、の 2 つの点を特徴とする。これらの特徴により、非同期通信をベースとしたシステムの上で高精度な同期ストリーム処理を実現することをめざしている。

現在、ISDN を用いた周波数/時刻同期システムが稼働中である。今後は本手法で用いた各種制御パラメータの測定方法の検証、詳細設計、製作、実アプリケーションへの適用、評価と進めていく。

謝辞

本研究をご支援下さる後藤滋樹広域コンピューティング研究部長をはじめ、同部の皆様に感謝いたします。

参考文献

- [1] 小倉毅, 小野諭, 高橋直久, "非同期な並列分散環境における実時間連続メディア処理", 信学技報, Vol.94, No.573, pp.33-40.
- [2] 葉原耕平, 井上伸雄, "デジタル総合網", 産業図書, 1989.
- [3] R.P.Singh, Sang-Hoon Lee, Chong-Kwoon Kim, "Jitter and Clock Recovery for Periodic Traffic in Broadband Packet Networks", IEEE trans.on communications, Vol.42, No.5, MAY 1994.
- [4] 藤原 洋 監修, マルチメディア通信研究会編, "最新 MPEG 教科書", アスキー, 1994.
- [5] 山下高生, 小野諭, "高速デジタル網を用いたクロック周波数同期", 信学技報, Vol.94, No.573, pp.25-32.
- [6] 山下高生, 小野諭, "ISDN 網を用いた分散高精度時刻/周波数同期", 情処研報, Vol.95, No.61, pp.37-42.