

# クロストラヒックの変動を考慮した回線エミュレーションにおける適応クロック法の検証

2004MT041 川越 崇史 2006MI129 小川 佳介 2006MI169 舘 滋之

指導教員 奥村 康行

## 1. はじめに

回線エミュレーションは、同期通信サービスを非同期通信ネットワーク上に構築する方式で、既に複数の方式が提案され標準化もされつつある。それによって、既存同期ネットワークと非同期ネットワークの統合が可能となり、ネットワークの経済化・柔軟化をもたらすことが期待できる。

回線エミュレーションでは、非同期通信ネットワーク上に構築される同期通信サービスの品質保持・劣化抑圧が技術的課題となる。この内、端末へのクロックの配信・同期は最も重要な技術的課題の1つであり、この課題の一般的な解決手段として適応クロック法がある[1]。

先行研究では、受信側のクロック回路に到着するパケットの時間間隔が、クロストラヒックの影響を受け一定ではないという予想下で研究が行われた[2]。この時、クロストラヒックとしてTCPを用いたもののみを考慮に入れたが、実際にはUDPを用いたクロストラヒックも存在し、パケットの到着時間間隔はさらに変動するものと予想される。本研究は受信側のクロック回路に到着するパケットの時間間隔の変動を分析する。そして、その結果に基づき適応クロック法を用いた最適なクロック回路の設計を行う。

## 2. 回線エミュレーションと課題

### 2.1 回線エミュレーションについて [1]

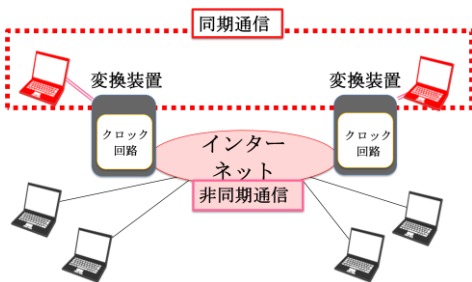


図1 回線エミュレーションの構成

回線エミュレーションとは、ユーザ側の電気通信端末に対して擬似的に電気通信事業者の網に接続されているように見せかける機能である。

図1では同期通信サービスを非同期通信ネットワークを介して接続する構成を示している。

同期通信とは、送信側と受信側でクロックの周波数を一致させる通信であり、専用線で採用されている。一方、インターネットは非同期通信ネットワークである。

### 2.2 適応クロック法

図1のクロック回路に適用される方式として適応クロック法がある。適応クロック法とは非同期ネットワークを介して受信端末に同期させる技術である。具体的には、受信側のクロック回路のバッファ蓄積量によって、再生周波数を決定する方式である。本研究では、適応クロック法としてPID制御とバッファ残量擬似平均方式の2つの制御方法を挙げる。

### 2.3 PID制御 [3]

PID制御の基本公式は式(1)で基本原理は3つの動作から成る。式(1)の $C_p(b(t) - b_0)$ はP制御、 $C_i \int (db(t)/dt)$ はI制御、 $C_d(db(t)/dt)$ はD制御である。

[基本公式]

$$f(t) - f(0) =$$

$$C_p(b(t) - b_0) + C_i \int (b(t') - b_0) dt' + C_d \left( \frac{db(t)}{dt} \right) \quad (1)$$

$f(t)$  :  $t$ の時間の周波数  $f(0)$  : 初期の周波数

$b(t)$  :  $t$ の時間のバッファ占有度

$C_p$  : 比例係数  $C_d$  : 微分係数  $C_i$  : 積分係数

P制御は現在の偏差に比例した修正量を出す比例制御(P制御)である。Iは過去の偏差の累積値に比例した修正量を出す積分制御(I制御)である。Dは偏差が増加しつつあるか減少しつつあるか、その傾向の大きさに比例した修正量を出す微分制御(D制御)である。

この3つを加算合成したものであり、バッファ残量に応じて目的の周波数に近似するように制御する。

## 2.4 バッファ残量擬似平均方式 [4]

バッファ残量擬似平均方式は過去 $N$ 回分のバッファ残量の擬似的な平均値を求め、その平均値に重み係数を掛けた値を制御値とするフィードバック制御である。

下記の式(2)で、サンプリングサイクル毎に過去 $N$ 回分の擬似平均値  $H_{S_n}$  を求めている。式(3)で、式(2)で求めた擬似平均値  $H_{S_n}$  に重み係数である感度 $A$ を掛けて制御値 $U_n$ を算出している。

[基本公式]

$$H_{S_n} = ((N - 1) * H_{S_{n-1}} + H_n) / N \quad (2)$$

$$U_n = A * H_{S_n} \quad (3)$$

$U_n$  : (n回目の) 制御値     $A$  : 感度     $N$  : 平均母数

$H_n$  : (n回目の) 観測値     $H_{S_n}$  : (n回目の) 平均値

## 3. ネットワークシミュレーションによるパ

### ケット到着時間間隔の解析

#### 3.1 ネットワークモデル [2]

ネットワークシミュレーションで受信側のクロック回路に到着するパケットの時間間隔の時刻に伴う変化を分析する。そこで図2のようなネットワークポロジを検討する。図2において、ノードn0とノードn1間とノードn2とノードn3間が専用線、それ以外のホストはインターネットとする。図2のn0-n3間のプロトコルはUDPを用い、スループットは64Kbpsに固定する。ノードn4-n23・ノードn24-n43間のプロトコルはTCPを、ノードn44-n45間のプロトコルはUDPを用いる。クロストラヒックにおけるUDPのスループットを1Mbps, 3Mbps, 5Mbpsと変化させた場合をモデルa, b, cとする。

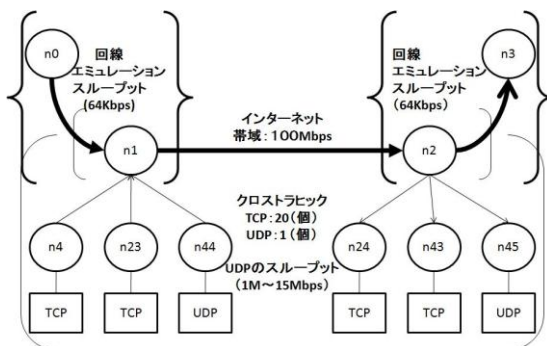


図2 ネットワークポロジ (TCP: 20 個, UDP: 1 個)

#### 3.2 時間間隔の変化

図2のネットワークポロジでシミュレーションを行う。シミュレーションは120秒間行い、回線エミュレーションはシミュレーション開始後直ちにパケットの送受信を開始する。また、開始後5秒から120秒まで

TCPのクロストラヒックが生じ、開始後10秒から60秒までUDPのクロストラヒックが生じる。このとき、パケットが受信側のクロック回路に到着する時間間隔を求め、その分布を図3に示す。

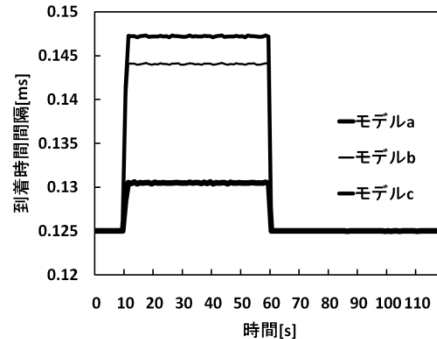


図3. 1秒毎のパケットの到着時間間隔の平均値の変化

#### 3.3 測定結果に対する考察

シミュレーション開始10秒後に到着時間間隔が大幅に増加し、シミュレーション開始60秒後に到着時間間隔が減少している。シミュレーション開始10秒後から60秒後におけるパケットの到着時間の平均値は、モデルa, b, cにおいて、0.131[ms], 0.144[ms], 0.147[ms]となる。以上のことから、TCPのクロストラヒックとUDPのクロストラヒックが同時に掛かった際に回線に大きな負荷が掛かったと考えられる。またUDPのクロストラヒックのスループットを大きくすると、回線に掛かる負荷も大きくなると考えられる。

## 4. 数値シミュレーションの結果

PID制御とバッファ残量擬似平均方式を用いた数値シミュレーションの結果を述べる。

#### 4.1 数値シミュレーションにおける収束・ジッタの定義 [2]

数値シミュレーションにおける収束時間は、受信側のクロック回路に到着するパケットの時間間隔が一定ではないため、目標周波数との誤差が±200ppm(12.8Hz)以内で収まったとき収束とみなす。

また、ジッタの定義は収束の定義により収束時のジッタ値を測定する。ジッタとは、クロック信号のようなパルス信号の位置や幅がずれたときのずれの量を指す。本研究においては式4でジッタを定義する。

[基本公式]

$$\text{jitter}[\text{ppm}] = \frac{f_{\max}[\text{Hz}] - f_{\min}[\text{Hz}]}{f} \times 10^6 \quad (4)$$

$f_{\max}, f_{\min}$  : 最大, 最小周波数     $f$  : 目標周波数[Hz]

## 4.2 PID制御

モデル a について具体的なパラメータを用いて PID 制御の数値シミュレーションを行う。送信信号はネットワークシミュレーションから得られた受信側のクロック回路に到着するパケットの時間間隔をそのまま用いる。

初期周波数を  $f_0 = 63,986.423518[\text{Hz}]$  として、目標の周波数を  $f = 64000[\text{Hz}]$  目標のバッファ占有度  $30[\text{Byte}]$ 、微分係数  $C_d = 0$ 、バッファ使用量のサンプリングサイクル  $T = 1[\text{ms}]$  としている。このとき数値シミュレーションでは、クロック再生回路がパケットを受信し始めてからの再生クロック周波数およびバッファ使用量の時間変化を求めた、PID 制御における数値シミュレーションは、比例係数  $C_p = 5000$ 、積分係数  $C_i = 100$  とし、結果を図 4 でグラフに示す。

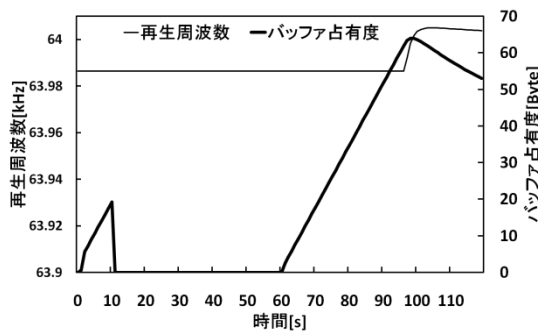


図 4 数値シミュレーションの結果 ( $C_p = 5000$ ,  $C_i = 100$  の例)

図 4 より、バッファ占有度は UDP のクロストラヒックが生じる 10 秒から 60 秒の間では 0[Byte] で、それ以降は目標のバッファ占有度に近似するように変化している。再生周波数は 90 秒までは変化せず、それ以降は目標の再生周波数に近似するように変化している。これについては以下の理由が考えられる。

PID 制御はバッファ占有度の増減に応じて再生周波数の制御を行っており、目標値よりバッファ占有度が小さい場合は再生周波数を下げる制御を行い、目標値よりバッファ占有度が大きい場合はバッファ占有度の増減に応じて再生周波数を上下させる制御を行う。またプログラム内で再生周波数を目標値に近似させる目的で、再生周波数が再生周波数の初期値  $f_0$  より小さい場合は、再生周波数を初期値  $f_0$  とする制御を行っている。

そのため 10 秒～90 秒において再生周波数は周波数の初期値に制御され、90 秒以降に再生周波数を上げる制御が行われ、100 秒以降に再生周波数を下げる制御が行われた。このとき再生周波数は 96.915 秒で収束する。

また、モデル b, c について数値シミュレーションを行った場合も同様の傾向が見られ、再生周波数の収束時間はそれぞれ 96.991 秒、96.997 秒となった。

次にジッタも観点に入れて、数値シミュレーションを行う。表 1 は、PID 制御における収束時間が 120 秒以内に収束する比例係数、積分係数のモデル (1, 2, 3, 4, 5) を示す。図 5 に数値パラメータ (表 1) における数値シミュレーションを行った結果を示す。

表 1 数値シミュレーションのモデル (1, 2, 3, 4, 5)

モデル	$C_p$	$C_i$
モデル 1	7000	80
モデル 2	6000	90
モデル 3	5000	100
モデル 4	4000	110
モデル 5	3000	120

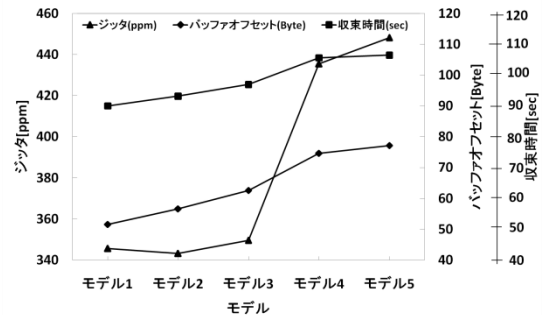


図 5 PID 制御(ジッタ, バッファオフセット, 収束時間)

図 5 より、ジッタについてはモデル 1 で 345.6 [ppm]、モデル 2 で 343.1 [ppm] となりほとんど変化がない。収束時間についてはモデル 1 で 90.0 [s] となり最短となる。またバッファオフセットについてもモデル 1 で 51.6 [Byte] となり最小となる。

よって結果として、ジッタが小さく、収束時間が最短、バッファオフセットが最小となるモデル 1 が適している。

## 4.3 バッファ残量擬似平均方式

モデル a について具体的なパラメータを用いてバッファ残量擬似平均方式の数値シミュレーションを行う。送信信号はネットワークシミュレーションから得られた受信側のクロック回路に到着するパケットの時間間隔をそのまま用いる。

初期周波数を  $f_0 = 63,986.423518[\text{Hz}]$  として、目標周波数を  $f = 64000[\text{Hz}]$ 、目標のバッファ占有度を  $4[\text{Byte}]$ 、サンプリングサイクル  $T = 1[\text{ms}]$  としている。平均母数  $N: 4096$  とする。バッファ残量擬似平均方式における数値シミュレーションについては感度  $A = 128$  で行い、図 6 のようなグラフの結果を得た。

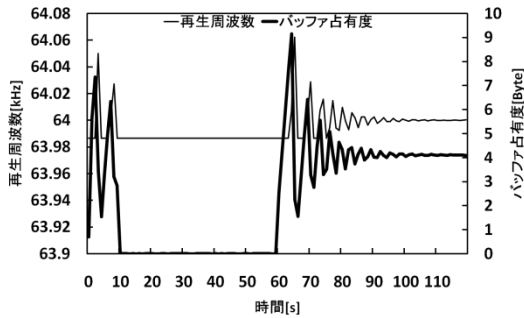


図6 数値シミュレーションの結果(感度A=128の例)

図6より、バッファ占有度はUDPのクロストラヒックが生じる10秒から60の間では0[Byte]で、それ以降目標のバッファ占有度に近似するように変化している。再生周波数は10秒から60秒の間では変化せず、それ以降は目標の再生周波数に近似するように変化している。これについては以下の理由が考えられる。

バッファ残量擬似平均方式では、再生周波数の初期値 $f_0$ に制御値 $U_n$ を足して再生周波数 $f_t$ とする制御を行う。バッファ占有度の目標値よりバッファ占有度が大きい場合は $U_n$ が正の値となる。また目標値よりバッファ占有度が小さい場合は $U_n$ は負の値となるが、再生周波数を目標値に近似させる目的で、 $U_n$ を0にする制御を行う。

図6において、10秒から60秒においてはバッファ占有度が目標値より常に小さいため、 $U_n$ は0となり再生周波数は初期値 $f_0$ となる制御が行われた。60秒以降はバッファ占有度は目標値より大きい値に収束していくため、 $U_n$ の値は正のある値となり再生周波数は $f_0$ より大きい値に収束する。よって10秒から60秒の間と90秒以降における再生周波数の収束値が異なる。このとき78.211秒で再生周波数はで収束する。また、モデルb,cについて数値シミュレーションを行った場合も同様の傾向が見られ、再生周波数の収束時間はそれぞれ78.479秒、78.481秒となった。

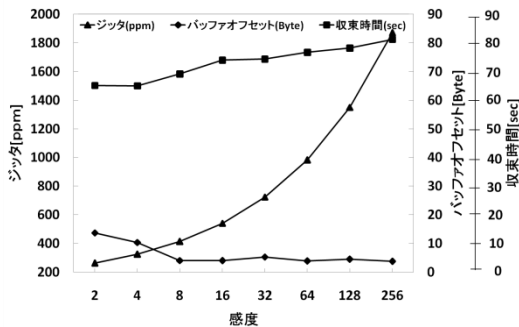


図7 バッファ残量擬似平均方式(ジッタ、バッファオフセット、収束時間)

次にジッタ・収束時間も観点に入れて、数値シミュレーションを行う。図7にバッファ残量擬似平均方式における感度A(2, 4, 8, 16, 32, 64, 128, 256)を変化させて、収束時間が常に120sec以内に収束した結果を示す。

図7より、感度Aが2のとき、ジッタ・収束時間は262.6[ppm], 65.2[s]となり最小、最短となる。バッファオフセットについては感度Aが256のときに3.7[Byte]となり最小となる。

よって結果として、ジッタ・収束時間を優先して考えた場合、ジッタ・収束時間が最小、最短となる感度A=2のときが適している。

## 5. まとめと今後の課題

本論文では、回線エミュレーションにおける適応クロック法として、PID制御とバッファ残量擬似平均方式の2つの制御方法を用いた時の有効性をバッファオフセット、ジッタ、収束時間の3つの観点から数値シミュレーションを行い、定性的・定量的に検証した。

その結果、バッファオフセットの大きさ、ジッタの安定性、収束するまでの時間を考慮すると、それらがより小さく、短くなるバッファ残量擬似平均方式が適応クロック法として実用的であるということが分かった。

今後の課題として、ネットワークシミュレーションによって、受信側のクロック回路に到着するパケットの到着時間間隔の時刻に伴う変化を分析するとき、さらにノードの数を増やし、スループットの組み合わせを変化させ、複雑なネットワークトポロジーを用いて、長時間のネットワークシミュレーションを行う必要がある。

また、PID制御、バッファ残量擬似平均方式のクロック再生回路について構成、試作、動作を検証し、コスト面から本方式がデジタル回路で実現されることを実証する必要がある。

## 参考文献

- [1] 舟橋拓郎, 伊藤寛和, 乗本清隆, “回線エミュレーションにおける適応クロックの比較と検証” 2008年度南山大学卒業論文, 2009.
- [2] 山田佳美, 波多野友香, “回線エミュレーションにおいてクロストラフィックを考慮した適応クロック法の比較と検証,” 2009年度南山大学卒業論文, 2010.
- [3] 深田陽一, 安田武, 小松秀司, 斎藤幸一, 前田洋一, “回線エミュレーション向け比例・積分・微分(PID)制御型適応クロック法の実装と評価” 電子情報通信学会, 信学技報, CS2005-43 (2005-11).
- [4] 村上謙, 横谷哲也, “TDM over Ethernetにおけるクロック精度検証,” 電子情報通信学会, 信学技法, CS2006-1, (2006-4).