社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
CQ2010-76(2011-03)

相互制御機構に基づく端末間同期方式

吉田　裕志† 　大芝　崇† 　大西　健夫† 　里田　浩三†

† NEC サービスプラットフォーム研究所　〒 211–8666 神奈川県川崎市中原区下沼部 1753
E-mail: †h-yoshida@jh.jp.nec.com, oshiba@cp.jp.nec.com, t-onishi@cj.jp.nec.com, k-satoda@cb.jp.nec.com

あらまし　複数ユーザがそれぞれの端末で同じ動画を同時に視聴しながら通話やチャットなどのコミュニケーションを行う動画共有視聴サービスでは，ネットワーク遅延によって同時刻に各端末で表示されている動画のシーンにずれが生じること (端末間同期外れ) が課題となる．端末間同期外れが大きいと，動画のシーンに対する反応に時間差が生じ，コミュニケーションが円滑に行うことができない．この課題に対し，これまでに数多くの端末間同期方式に関する研究がなされてきた．しかし，その多くが高精度なグローバルクロックを前提にした方式や，ネットワーク遅延についての統計情報を予め要する方式であった．本稿では，同期制御サーバからだけでなく，端末間でも互いに同期制御信号をやりとりすることで，グローバルクロックやネットワーク遅延の前提条件なしに，同期精度を向上させる端末間同期方式を提案する．シミュレーションにより，提案方式は，従来方式と比較して倍以上の同期精度を達成できる上に，視聴端末数が増加するほど同期精度が向上することを示す．

キーワード　端末間同期，相互制御，ネットワーク遅延，動画共有視聴，再生制御

An Inter-Terminal Synchronization Scheme based on a Mutual Control Mechanism

Hiroshi YOSHDIA†, Takashi OSHIBA†, Takeo ONISHI†, and Kozo SATODA†

† Service Platforms Res. Labs., NEC Corp.　1753 Shimonumabe, Nakahara-ku, Kawasaki, 211–8666 Japan
E-mail: †h-yoshida@jh.jp.nec.com, oshiba@cp.jp.nec.com, t-onishi@cj.jp.nec.com, k-satoda@cb.jp.nec.com

Abstract　Scene synchronization is important for a video sharing service where users can call or chat watching the same video content. A wide synchronization gap, which is caused by a network delay, creates a time lag among users' reaction to the scenes, and spoils communication quality among them. A number of studies about an inter-terminal synchronization scheme have been reported; however, most of them are based on a high accuracy global clock or given statistical information about the network delay. In this paper, we propose an inter-terminal synchronization scheme in which synchronization control signals are exchanged among not only server-terminal but also terminal-terminal. Simulation results show that our proposed scheme achieves more than double-accuracy of synchronization compared with a conventional one, and the more terminals, the more accurately they can synchronize.

Key words　Inter-Terminal Synchronization, Mutual Control, Network Delay, Video Sharing, Playout Control

1. はじめに

ネットワークのブロードバンド化と通信端末の高性能化に伴い，動画ストリーミングサービスが急速に発展してきた．今後は，単に動画を観るだけでなく，複数ユーザがそれぞれの端末で同じ動画を同時に視聴しながら通話やチャットなどのコミュニケーションを行う動画共有視聴サービスが，特にモバイル通信の分野で普及してくると筆者らは予想している．モバイル通信の業界団体である GSMA が現在規定している RCS (Rich Communication Suite) の主機能のひとつである Multimedia Sharing の代表例に，この動画共有視聴が挙げられる [1] ことも，この予想を裏付けている．

複数ユーザがそれぞれの端末で同じ動画を同時に視聴しながら会話をする場合，同時刻に各端末で表示されている動画のシーンにずれが生じることが課題となる．これを端末間同期外れと呼ぶ．端末間同期外れが大きいと，動画のシーンに対する反応に時間差が生じ，コミュニケーションが円滑に行うことができない．端末間同期外れの主な原因は，ネットワーク遅延にある．ネットワーク遅延の違いにより，動画の同期制御のための信号が各端末へ到達する時刻にずれが発生し，端末間同期外れとなる．

この端末間同期外れの問題に対しては，数多くの従来研究がある [2]–[17]．しかしながら，その多くが高精度なグローバルク

ロックを前提にした方式や，ネットワーク遅延について既知情報を要する方式であり，種々の端末やネットワークを対象にしたサービスでは実用性に欠ける．

また，モバイルネットワークにはモバイル端末–基地局間で自動的に時刻同期する機能が備わっているものがあるが，ヘテロジニアスなネットワーク環境では各端末間の時刻同期は保証されない．

そこで，本稿では，同期制御サーバからだけでなく，端末間でも互いに同期制御信号をやりとりすることで，グローバルクロックやネットワーク遅延の前提条件なしに，同期精度を向上させる端末間同期方式を提案する．本稿の構成は次のとおりである．2 章では，端末間同期外れの問題について詳しく説明するとともに，従来の同期方式について分類・整理する．3 章では，提案する同期方式について詳細を述べ，その同期精度を数学的に解析する．4 章では，実網でのネットワーク遅延データに基づいたシミュレーションにより，提案方式と従来方式の同期精度を比較した．そして，5 章で，本稿のまとめと，今後の研究課題について述べた．

2. 端末間同期外れと従来技術

2.1 端末間同期外れ

複数ユーザがそれぞれの端末で同じコンテンツを同時に視聴する際，各端末で同時刻に表示もしくは再生されてるコンテンツの内容にずれが生じることを端末間同期外れという．コンテンツとしては動画，音声，テキスト等のマルチメディアデータがあるが，本稿では動画を例にとる．

端末間同期外れは，同期の「外れ方」によって「初期同期外れ」と「途中同期外れ」に分類できる．

初期同期外れは，同期開始時 (例えば，動画再生開始時) に発生する同期外れである．初期同期外れの原因は，ネットワーク遅延の違いによって，各端末における動画の再生制御信号の受信時刻に差が生じることにある．同期制御サーバから各端末に向けて同時刻に再生制御信号を送信しても，上述の受信時刻の差により，各端末における動画再生開始時刻に差が生まれる．

一方，途中同期外れは，同期開始時点では同期外れがなくとも，途中から徐々に，もしくは急激に同期外れが発生することである．途中同期外れの原因はふたつある．ひとつめの原因は，端末間のクロックスキューである．端末間のクロックスキューによって，徐々に再生位置がずれていくことで同期外れとなる．ふたつめの原因は，動画をストリーミングしながら視聴している場合，ネットワーク帯域不足によって再生バッファが枯渇し，動画再生が途絶してしまうことである．一方の端末の動画再生が途絶してしまうと，そこで大きな同期外れが発生することになる．

本稿では，ネットワーク遅延が原因の初期同期外れを対象にする．途中同期外れの問題については，別途検討する．

2.2 従来の端末間同期方式

上述した端末間同期外れの課題に対して，これまでに多くの研究がなされてきた [2]–[17]．筆者らは，これらの従来研究をふたつの軸で分類した (表 1)．

ひとつめの分類軸は，離れた場所に位置する端末間の時刻同期をとる方法 (時刻同期方法) についての分類である．時刻同期方法で分類すると，グローバルクロック方式と $\frac{1}{2}$RTT 方式 (片道遅延推定方式) がある．グローバルクロック方式は，高精度かつ全端末から参照可能なグローバルクロックを用意することで時刻の同期をとる方式である．グローバルクロックとしては NTP や GPS クロックが挙げられている．一方，$\frac{1}{2}$RTT 方式 (片道遅延推定方式) は，グローバルクロックを用いずに同期をとる方式である．あるひとつの装置 (ここでは制御サーバとする) が有する時計を基準として，各端末に時刻を通知する．このとき，制御サーバ–端末間のネットワーク片道遅延量だけ時刻通知が遅れてしまうため，その分だけ進んだ時刻を端末に通知しなければならない．ただし，ネットワーク片道遅延量は End-to-End だけでは計測できないため，End-to-End だけで計測可能な RTT(往復遅延量) の $\frac{1}{2}$ で代用する．すなわち，制御サーバが端末に時刻を通知する場合，通知する時点の時刻を t とすると，$t+\frac{1}{2}$RTT を端末に通知することになる．従来技術は，前者のグローバルクロック方式を採用したものが大部分である [2]–[16]．

表 1　端末間同期方式の分類

＜時刻同期方法での分類＞	＜制御方法での分類＞
・グローバルクロック方式	・マスタ・スレーブ端末方式
・$\frac{1}{2}$RTT 方式 (片道遅延推定方式)	・同期マエストロ方式
	・分散制御方式

ふたつめの分類軸は，同期のための制御信号をやりとりする方法 (制御方法) についての分類である．制御方法の分類については [11] に詳しく記載されており，マスタ・スレーブ端末方式 [4]，同期マエストロ方式 [6]，分散制御方式 [7] の 3 方式がある．マスタ・スレーブ端末方式は，端末の 1 台がマスタ端末として同期タイミングをとる決定権を有し，その他の端末はスレーブ端末としてマスタ端末からの指示に従う方式である．同期マエストロ方式は，ネットワーク上に同期制御を司る同期マエストロを設ける方式である．同期マエストロは，各端末から同期のためのタイミング情報を収集し，基準とすべき同期タイミングを決定した後に，各端末へ同期制御信号を通知する．分散制御方式では，上記ふたつの方式と異なり同期タイミングを集中管理する装置を設けず，各端末が他端末から受信したタイミング情報に基づいて同期タイミングをそれぞれ決定する．

3. 端末間相互制御方式

筆者らは，複数人がそれぞれの端末で同じ動画を同時に視聴しながらコミュニケーションをとる動画共有試聴サービスにおける動画同期を主な研究対象としている．近年の通信端末及びネットワークは多様化しており，全ての端末で参照可能な高精度なグローバルクロックを用いることは実用性に欠く．グローバルクロックを前提としない場合，特にネットワーク遅延量が大きいモバイルネットワークでは，同期精度の改善が必要となる．また，複数人で動画共有試聴する場合，端末数が多くなればなる程，同期外れが大きい端末が存在する確率が高くなってしまう問題もある．

そこで，本稿では，グローバルクロックを用いず，かつ従来の制御方式よりも高精度な同期制御方式を提案する．加えて，端末数が多くなった場合でも同期精度が低下しないことを目標とする．

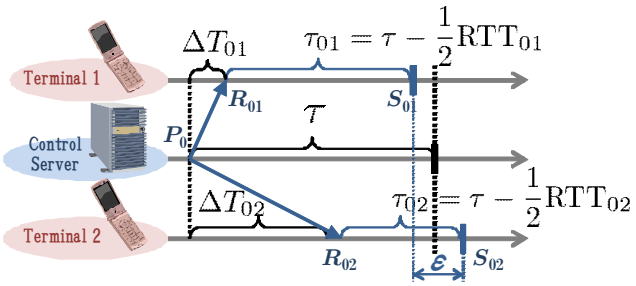


図 1　従来方式 ($\frac{1}{2}$RTT 方式×同期マエストロ方式)

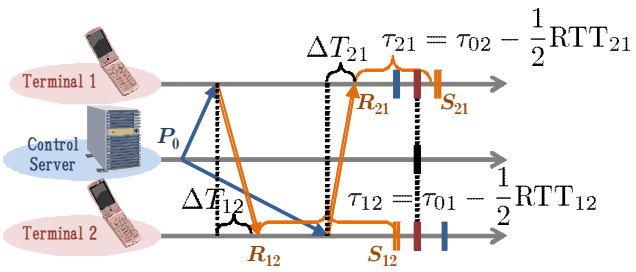


図 2　提案方式 ($\frac{1}{2}$RTT 方式×相互制御方式)

3.1　端末間相互制御方式の仕組み

本稿で提案する同期制御方式は，同期制御サーバからだけでなく，端末間でも互いに同期制御信号をやりとりすることで，グローバルクロックやネットワーク遅延の前提条件なしに，同期精度を向上させる方式である．図 1 及び図 2 を参照し，提案方式の仕組みについて従来方式と比較して詳細に説明する．以降の説明では端末数を n に一般化した場合を考えるが，図では簡略化のため端末数が 2 の場合を示している．

図 1 は，従来方式の代表として，時刻同期方法に $\frac{1}{2}$RTT 方式，制御方法に同期マエストロ方式 (もしくはマスタ・スレーブ端末方式) を採用したときの同期の仕組みを示した図である．制御サーバ及び端末から右に延びている矢印は時間軸であり，ある時間軸上から他の時間軸上へ伸びている斜め方向の矢印 (例えば $P_0 \rightarrow R_{01}$) が同期制御信号を示している．この図は，制御サーバが P_0 の時点から時間 τ 後に同期しようとするときの制御信号の内容を表している．制御サーバは，同期制御信号として端末 $i(i = 1, 2, \cdots, n)$ に対して同期待ち時間を通知する．このとき，制御サーバから端末 i に通知する同期待ち時間 τ_{0i} は式 (1) になる．

$$\tau_{0i} = \tau - \frac{1}{2}\mathrm{RTT}_{0i} \tag{1}$$

ただし，RTT_{0i} は制御サーバ–端末 i 間の RTT である．端末 i が上記同期制御信号を受信した時刻を R_{0i} とすると，端末 i での同期時刻 S_{0i} は式 (2) と書ける．

$$S_{0i} = R_{0i} + \tau_{0i} \tag{2}$$

もし，制御サーバから端末 i への片道遅延量 ΔT_{0i} が $\frac{1}{2}\mathrm{RTT}_{0i}$ に等しければ正確に同期可能だが，実際には制御サーバから端末 i への片道遅延量と端末 i から制御サーバへの片道遅延量は等しくならないため，端末間の同期時刻にずれ (図 1 の ε) が発生する．このずれは次節で定量的に解析する．

一方，提案方式の仕組みは図 2 のようになる．従来方式との違いは，各端末が制御サーバからの同期制御信号を受信した後，即時に他方の端末に対して同期制御信号を通知している点にある．各端末が送信する同期制御信号も，制御サーバのそれと同様である．制御サーバから通知された同期待ち時間 τ_{0i} に基づいて，他の端末 $j(j = 1, 2)$ へ次式 (3) で表される同期待ち時間 τ_{ij} を通知する．

$$\tau_{ij} = \tau_{0i} - \frac{1}{2}\mathrm{RTT}_{ij} \quad (i \neq j) \tag{3}$$

ただし，RTT_{ij} は端末 i–端末 j 間の RTT である．端末 j では，上記同期待ち時間 τ_{ij} の通知を受けて，この同期待ち時間に基づいた同期時刻 S_{ij} を式 (4) で計算できる．

$$S_{ij} = R_{ij} + \tau_{ij} \quad (i \neq j) \tag{4}$$

ただし，R_{ij} は上記同期待ち時間通知の受信時刻である．このように，制御サーバからの同期制御信号だけでなく，端末間でも同期制御信号をやりとりすることで，各端末は，制御サーバ及び他の端末からの複数の同期時刻情報 (制御サーバから S_{0i}，他端末から S_{ji}) を得ることになる．

これら複数の同期時刻情報から，最終的な同期時刻を決定する．端末 i における同期時刻 S_i の最も簡易な決定方法は，式 (5) のように複数の同期時刻の平均値を採用した．

$$S_i = \frac{1}{n}\sum_{j=0, j\neq i}^{n} S_{ji} \tag{5}$$

このように，複数経路から同期制御信号を受信して同期時刻を決定することで，ネットワークの片道遅延量を $\frac{1}{2}$RTT で推定したときの誤差を小さくすることが可能になる．誤差を小さくできる理由については，次節 3.2 で詳しく述べる．

3.2　同期精度の解析

提案する同期制御方式は，同期制御サーバからだけでなく，端末間でも互いに同期制御信号をやりとりして同期時刻を決定する方法であることを説明した．本節では，提案方式によって同期精度が向上する理由について述べる．

制御サーバが同期制御信号を送信する時刻を t とする．このとき，端末 i での同期制御信号の受信時刻 R_{0i} は式 (6) で書ける．

$$R_{0i} = t + \Delta T_{0i} \tag{6}$$

ただし，ΔT_{0i} は，制御サーバから端末 i までの片道遅延量である．式 (1) と式 (6) を式 (2) に代入すると，制御サーバからの同期時刻 S_{0i} は式 (8) のようになる．

$$S_{0i} = t + \tau + \Delta T_{0i} - \frac{1}{2}RTT_{0i} \tag{7}$$

$$= t + \tau + \frac{\Delta T_{0i} - \Delta T_{i0}}{2} \tag{8}$$

式 (7) から式 (8) への変形は，$RTT_{0i} = \Delta T_{0i} + \Delta T_{i0}$ であることを用いた．式 (8) の右辺第 3 項 $\frac{\Delta T_{0i} - \Delta T_{i0}}{2}$ が従来方式における制御サーバが設定した同期時刻との誤差に相当する．

一方，端末 i が端末 j から同期制御信号を受け取るとき，同期制御信号は制御サーバ→端末 j →端末 i という経路を辿るので，端末 i での受信時刻 R_{ji} は式 (9) のようになる．

$$R_{ji} = t + \Delta T_{0j} + \Delta T_{ji} \tag{9}$$

ここで，式 (4) に式 (1)，(3)，(9) を代入すると，端末 j から通知された同期時刻 S_{ji} は次式 (11) で書ける．

$$S_{ji} = t + \tau + \Delta T_{0j} - \frac{1}{2}RTT_{0j} + \Delta T_{ji} - \frac{1}{2}RTT_{ji} \quad (10)$$

$$= t + \tau + \frac{\Delta T_{0j} - \Delta T_{j0}}{2} + \frac{\Delta T_{ji} - \Delta T_{ij}}{2} \quad (11)$$

端末 i における最終的な同期時刻 S_i は，式 (5) に式 (8)，(11) を代入すると，式 (12) のようになる．

$$S_i = t + \tau + \frac{1}{n}\sum_{j=1,j\neq i}^{n}\frac{\Delta T_{0j} - \Delta T_{j0}}{2} + \frac{1}{n}\sum_{j=0,j\neq i}^{n}\frac{\Delta T_{ji} - \Delta T_{ij}}{2} \quad (12)$$

式 (12) の右辺第 3 項と第 4 項が提案方式における制御サーバが設定した同期時刻との誤差に相当する．第 3 項及び第 4 項は，制御サーバと端末 j の間，及び端末 j と端末 i の間における往路遅延量と復路遅延量の差 (の半分) を平均した値である．あるひとつのネットワーク経路に着目すれば，往路遅延量と復路遅延量は異なる．しかし，複数経路における往路遅延量と復路遅延量の差を足し合わせて平均をとれば，経路のサンプル数が増加するにつれて，大数の法則によりネットワーク内の全経路における往路遅延量と復路遅延量の差の真の平均値に近づいていく．また，ある特定のネットワーク経路の往路遅延量と復路遅延量が非対称であったとしても，ネットワーク内の全経路の平均を考えるとほぼ対称と仮定できる．ゆえに，式 (12) の右辺第 3 項と第 4 項は n の増加に伴い 0 に漸近していく．すなわち，各端末における同期時刻 S_i は，制御サーバが設定した同期時刻 $t+\tau$ に近くなる．

一方，従来方式では，各端末で $\frac{\Delta T_{0i} - \Delta T_{i0}}{2}$ の同期時刻の誤差が発生する．そのため，端末数が増加するにつれて，確率的に端末間の誤差の開きが大きくなってしまう．

提案方式が従来方式と比較してどれだけ同期精度を改善できるかついては，4. でシミュレーションによる定量評価を実施する．このとき，端末数による同期精度の改善率の違いについても検証する．

3.3 加重平均による同期精度の向上

最終的な同期時刻 S_i を決定する際，式 (5) のように相加平均を用いた．しかし，RTT が小さなネットワーク経路を経由した同期制御信号で計算した同期時刻の方が，確率的に制御サーバが設定した同期時刻に近くなることは明らかである．そこで，相加平均ではなく，RTT が小さなネットワーク経路を経由した同期制御信号に重きを置く加重平均を用いることで，同期精度をより向上させる方法を提案する．具体的な計算方法は次のとおりである．

まず，時刻 $R_{ji}(j = 0, 1, \cdots, n,\ j \neq i)$ に受信した同期制御信号が経由したネットワーク経路の RTT を RTT_{ji}^{C} とする．制御サーバからの同期制御信号は直接通知されるので，$j = 0$ のときは $\mathrm{RTT}_{0i}^{C} = \mathrm{RTT}_{0i}$ である．一方，端末 j からの同期制御信号は制御サーバ→端末 j →端末 i と経由するので，$j \neq 0$ のときは $\mathrm{RTT}_{ji}^{C} = \mathrm{RTT}_{0j} + \mathrm{RTT}_{ij}$ である．

次に，この RTT_{ji}^{C} を用いて，加重平均の重み係数を求める．本稿では，式 (13) に示すように RTT_{ji}^{C} の 2 乗に反比例する値を採用することで，RTT が小さなネットワーク経路を経由した同期制御信号に大きな荷重がかかるようにする．

$$w_j = (\mathrm{RTT}_{ji}^{C})^{-2} \quad (13)$$

上記重み係数を用いて同期時刻 S_i は，式 (14) の加重平均で計算される．

$$S_i = \left(\sum_{j=0,j\neq i}^{n} w_j\right)^{-1}\left(\sum_{j=0,j\neq i}^{n} w_j S_{ji}\right) \quad (14)$$

4. のシミュレーション評価では，同期時刻 S_i に上記加重平均の値を用いた．

4. シミュレーション評価

4.1 評価項目

3.1 で説明した提案方式の同期精度についてシミュレーションによる定量評価を実施する．同期精度としては，「最大同期外れ $G_{\max}$」と「同期外れ標準偏差 G_{std}」の 2 項目を評価する．最大同期外れは，全端末の最早同期時刻と最遅同期時刻との差であり，同期外れの最悪値を評価する．同期外れ標準偏差は，全端末の同期時刻の標準偏差であり，同期時刻のばらつきを評価する．それぞれ，式 (15)，(16) で定義する．

$$G_{\max} = \max_{i=1\ldots n}\{S_i\} - \min_{i=1\ldots n}\{S_i\} \quad (15)$$

$$G_{\mathrm{std}} = \frac{1}{n}\sum_{i=1}^{n}(S_i - \bar{S})^2 \qquad \left(\bar{S} = \frac{1}{n}\sum_{i=1}^{n} S_i\right) \quad (16)$$

また，従来方式と比較して，提案方式がどれだけ最大同期外れと同期外れ標準偏差を改善できたかを評価するため，式 (17) で定義される「改善率 I」を計算する．

$$I = 1 - \frac{G^{\mathrm{p}}}{G^{\mathrm{c}}} \quad (17)$$

ただし，G^{p} は提案方式の最大同期外れ及び同期外れ標準偏差であり，G^{c} は従来方式のそれを指している．

4.2 シミュレーション条件

制御サーバ–端末間，及び端末–端末間の片道遅延量 ΔT_{ij} を後述する確率密度関数に従った乱数生成によって設定し，提案方式と従来方式で最大同期外れと同期外れ標準偏差を求める．

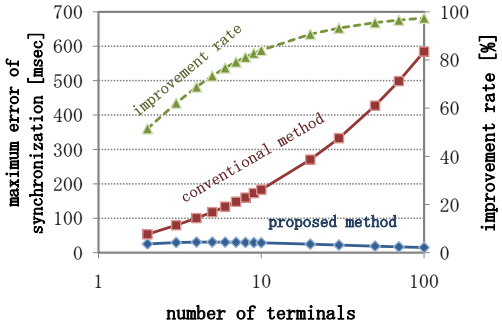
片道遅延量の確率密度関数には，式 (18) で表されるパレート分布を用いた．

$$f(x) = \frac{ab^a}{x^{a+1}} \qquad a > 2,\ b > 0 \quad (18)$$

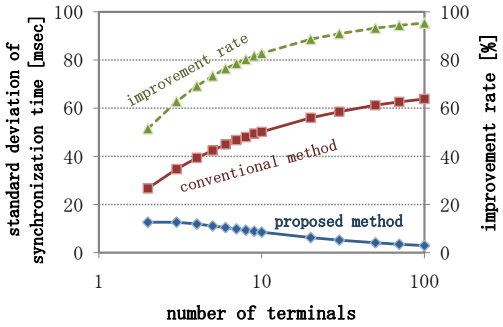
この理由は，同期外れが発生しやすいのは片道遅延量が大きくなりやすいネットワーク混雑時であり，混雑時の片道遅延量はパレート分布に従うと報告されているためである [18].

パレート分布のパラメータ a，b は，モバイル端末とインターネット上のサーバ間の RTT データから片道遅延量の平均値 (E=138.1[msec]) 及び標準偏差 (σ=111.4[msec]) を計算し(注1)，

(注1)：片道遅延量の平均及び分散を E_{one}，σ_{one} とすると，RTT の平均 E 及び分散 σ は，$E = 2E_{\mathrm{one}}$，$\sigma = \sqrt{2}\sigma_{\mathrm{one}}$ となる．これから，$E_{\mathrm{one}} = E/2$ 及び $\sigma_{\mathrm{one}} = \sigma/\sqrt{2}$ を導く．

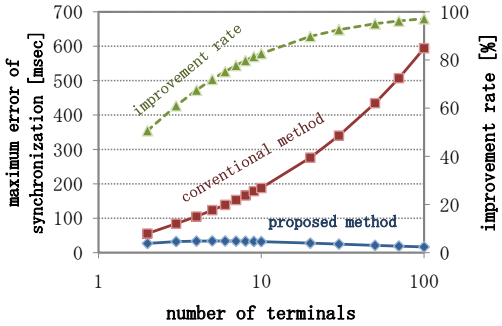


(a) 最大同期外れ

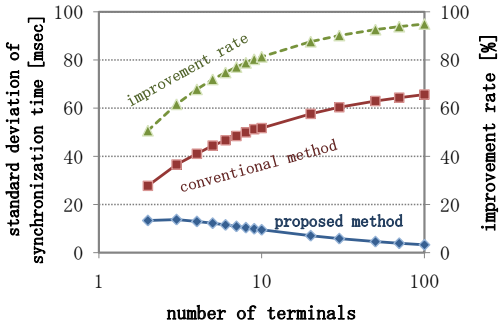


(b) 同期外れ標準偏差

図 3　シミュレーション結果 (RTT 計測誤差なし)

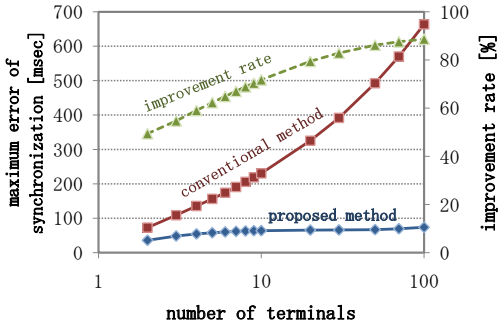


(a) 最大同期外れ

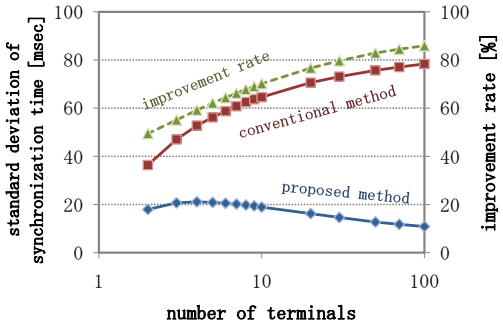


(b) 同期外れ標準偏差

図 4　シミュレーション結果 (RTT 計測誤差 10%)



(a) 最大同期外れ



(b) 同期外れ標準偏差

図 5　シミュレーション結果 (RTT 計測誤差 30%)

式 (19) に従って求めた．RTT データは，都内 12 箇所の実モバイルネットワーク経由で 158 回計測したものである．ただし，$\rho = \sqrt{1 + \left(\frac{E}{\sigma}\right)^2}$ である．

$$\begin{cases} E = \dfrac{ab}{a-1} \\ \sigma^2 = \dfrac{ab^2}{(a-1)^2(a-2)} \end{cases} \Rightarrow \begin{cases} a = 1 + \rho \\ b = \dfrac{E\rho}{1+\rho} \end{cases} \qquad (19)$$

端末数は n=2，3，4，5，6，7，8，9，10，20，30，50，70，100 の 14 通りで，それぞれの端末数で 10 万回のシミュレーションを実施した．その際，片道遅延量は上述のパレート分布に基づいて 1 回のシミュレーションごとに設定し直した．

ところで，実際問題として，RTT は時間的に変動するため，同期制御に用いた値 (事前に計測した RTT) と同期制御信号を送信した時点での値との間に誤差が生じる．そこで，計測した RTT に計測誤差が含まれる場合についても評価した．誤差のパターンとして，誤差なし，誤差が 10%程度のホワイトノイズ，誤差が 30%程度のホワイトノイズの 3 通りでシミュレーションを実施した．

4.3　シミュレーション結果

上述したシミュレーションの結果として，最大同期外れ，同期外れ標準偏差，及び改善率を図 3〜図 5 に示した．最大同期外れと同期外れ標準偏差は 10 万回のシミュレーション結果の平均値であり，改善率は 99%信頼区間の推定値である．図 3 は RTT の計測誤差がない場合，図 4 及び 5 はそれぞれ 10%及び 30%の計測誤差が含まれる場合である．

まず，最大同期外れの誤差なしの結果 (図 3(a)) について考察する．従来方式の最大同期外れは，端末数 2 のときに 53.4[msec] であり，以降端末数の増加に伴い急増し，端末数 100 では 584.7[msec] に達した．前述したとおり，端末が増加すれば同期外れが大きな端末の出現確率が増加するためである．一方，提案方式では，端末数 2 のときに 25.2[msec] であり，端末数 5 の 30.7[msec] まで

微増するが，以降は端末数 100 の 14.9[msec] まで最大同期外れを小さくすることができる．これは，端末数増加による最大同期外れ増加の傾向よりも，大数の法則による誤差の低減効果が強いことを示している．99%信頼区間の改善率は，端末数 2 のときに 51.5[%] となり，以降端末数に対して単調増加となり，端末数 100 では改善率 97.5[%] まで達成できた．

次に，同期外れ標準偏差の誤差なしの結果 (図 3(b)) を見る．従来方式は，端末数 2 のときに 26.7[msec] となり，端末数増加に伴い山形の増加を示し，端末数 100 では 95.4[msec] であった．提案方式は，端末数 2 のときに 12.6[msec] であり，端末数に対して単調減少を示し，端末数 100 では 2.9[msec] まで小さくすることできた．99%信頼区間の改善率は，端末数 2 のときに 51.5[%] となり，以降端末数に対して単調増加となり，端末数 100 では改善率 95.4[%] となった．

最大同期外れ，同期外れ標準偏差ともに，提案方式は改善率 50[%] 以上を達成できた．改善率 50[%] は，従来方式の同期外れを半減できたと同値であり，すなわち倍の同期精度を実現できたことを意味する．

最後に，RTT 計測誤差の影響について考察する．RTT の計測誤差が 10%の場合 (図 4) は，従来方式と提案方式ともにわずかに最大同期外れ及び同期外れ標準偏差が増加しただけであった．改善率についても，ほとんど変りなかった．計測誤差が 30%の場合 (図 5)，最大同期外れと同期外れ標準偏差の双方において，従来方式と提案方式ともに並べて見て分かる程度に同期外れが大きくなった．従来方式は，誤差が 0 のときのグラフを全体的に上昇させた形になった．一方，提案方式は，RTT 計測誤差により大数の法則による誤差の低減効果がやや薄れ，同期外れのピークが後方にずれた形になった．ただし，この場合でも同期精度は端末数 2 ではほぼ倍精度であり，端末数に対して単調増加を示した．

5. ま と め

本稿では，同期制御サーバからだけでなく，端末間でも互いに同期制御信号をやりとりすることで，グローバルクロックやネットワーク遅延の前提条件なしに，同期精度を向上させる端末間同期方式を提案した．シミュレーションにより，提案方式は，従来方式と比較して倍以上の同期精度を達成できる上に，視聴端末数が増加するほど同期精度が向上することを示した．

今後は，提案方式の負荷低減方法，提案方式の動画共有視聴システムへの適用について検討したい．

文　献

[1] GSM Association, “GSMA Rich Communication Suite White Paper,” http://www.gsmworld.com/our-work/mobile_lifestyle/rcs/index.htm, Oct. 2008.

[2] I. F. Akyildiz, W. Yen, “Multimedia Group Synchronization Protocols for Integrated Services Networks,” IEEE Journal on Selected Areas in Communications, Vol.14, No.1, pp.162-173, Jan. 1996.

[3] E. Biersack, W. Geyer, C. Bernhardt, “Intra- and Inter-Stream Synchronization for Sotred Multimedia Streams,” Proc. of IEEE Multimedia Computing and System, pp.372-381, 1996.

[4] Y. Ishibashi, A. Tsuji, S. Tasaka, “A Group Synchronization Mechanism for Stored Media in Multicast Communications,” Proc. of IEEE INFOCOM’97, Vol.2, pp.692-700, Apr. 1997.

[5] 辻 亮宏，石橋 豊，田坂 修二，“マルチキャスト通信におけるライブメディアに対する端末間同期方式,” 信学技報，IN96-130, pp.17-24，1997 年 2 月．

[6] Y. Ishibashi, S. Tasaka, “A Group Synchronization Mechanism for Live Media in Multicast Communications,” Proc. of IEEE GLOBECOM’97, pp.746-752, Nov. 1997.

[7] 石橋 豊，田坂 修二，中村 剛，“マルチキャスト通信におけるグループ同期：分散制御方式,” 信学技報，CQ98-48，pp.21-28，1998 年 10 月．

[8] 石橋 豊，田坂 修二，“ネットワーク環境における連続メディア同期技術,” 信学技報，SSE98-196，pp.61-66，1999 年 1 月．

[9] 田坂 修二，“ネットワーク環境におけるメディア同期,” 電子情報通信学会誌，Vol.84，No.3，pp.177-183，2001 年 3 月．

[10] 田坂 修二，石橋 豊，“分散マルチメディアアプリケーションにおける QoS と QoS マッピング,” 信学技報，CQ101，pp.7-14，2001 年 6 月．

[11] 布目 敏郎，田坂 修二，“中規模マルチキャスト通信における端末間同期方式のアプリケーションレベル QoS 比較,” 信学技報，CQ2002-86，pp.11-16，2002 年 9 月．

[12] 石橋 豊，田坂 修二，“分散マルチメディアアプリケーションにおけるメディアの時間構造と QoS,” 電子情報通信学会誌，Vol.87，pp.220-226，2004 年 3 月．

[13] T. Nunome, S. Tasaka, “Inter-Destination Synchronization Schemes for Continuous Media Multicasting: An Application-Level QoS Comparison in Hierarchical Networks,” IEICE Trans. Commun., Vol.E87-B, No.10. Oct. 2004.

[14] 田坂 和之，今井 尚樹，磯村 学，井戸上 彰，“FMC 環境下でのリアルタイムグループ通信におけるメディア同期方式,” 信学技報，IN2008-71，pp.23-28，2008 年 10 月．

[15] 高野 祐太郎，大島 浩太，寺田 松昭，“投稿型動画視聴におけるユーザ間リアルタイムコミュニケーション支援システムの提案,” 情報処理学会第 70 回全国大会，第 3 分冊，pp.351-352，2008.

[16] F. Boronat, M. Montagud, J. C. Guerri, “A New Netowork Simulatro 2 (NS-2) Module Based on RTP/RTCP Protocols to Achieve Multimedia Group Synchronization,” Proc. of the 3rd International ICST Conference on Simulation Tools and Techniques, 2010.

[17] 大西 健夫，城島 貴弘，中島 一彰，“コミュニケーション中の動画同期視聴を可能とする再生タイミング制御方式,” 情報処理学会第 72 回全国大会，2010.

[18] 藤本 康平，阿多 信吾，村田 正幸，“インターネットにおける実測に基づいたパケット伝送遅延の統計的分析,” 信学技報，IN-100(208)，pp.41-48，2000 年 7 月．