モバイルコンピューティングと
ワイヤレス通信 15−11
高度交通システム 3−11
（2000．12．1）

メンバ間公平性保証方式の同期機構の特性評価

石川 貴士* 石原 進*† 井手口 哲夫*† 水野 忠則‡

筆者らは，比較的即応性が要求されるネットワーク対戦ゲームにおいて，端末間の遅延差により起こるユーザ間の不公平さを解決するメンバ間公平性保証方式（*ICEGEM*: Impartial Communication Environment for GamE Members）を提案している．本方式では，反応時間に基づく公平性保証，クライアント側の制限時間，サーバ側の動的な制限時間・制限時間の短縮により，サーバ・クライアント間での時刻の同期，端末間の表示時刻の同期などの複雑で困難な制御を必要としないで，完全なユーザ間の公平性を実現する．本稿では，本方式を用い，ネットワーク対戦型早押しキーボードタイピングゲームを実装し，遅延差が生じる端末間で実験を行った．その結果，反応時間に基づく公平性保証では，端末間の遅延差に関係なく，公平にアプリケーションが利用できることが確認できた．また，サーバ側の動的な制限時間によって，変動する遅延に対応し，公平性を保証しながら即応性の向上を実現することができた．

Evaluation of Synchronous Mechanism in ICEGEM, with Guaranteed Fairness of Users' Response

Takashi Ishikawa*, Susumu Ishihara*†, Tetsuo Ideguchi*† and Tadanori Mizuno‡

In this paper we report the evaluation of the synchronous mechanism in *ICEGEM* (Impartial Communication Environment for GamE Members), which can control the order of the operations using users' response time. In real-time network applications, such as network games, the users often suffer event inversion on heterogeneous network environment which have small delay networks and large delay networks. Because of this, the fairness between users can not be guaranteed. We have proposed *ICEGEM* which guarantees the fairness of users' response time. We implemented a network key-typing game on *ICEGEM*, and evaluated Synchronous Mechanism in *ICEGEM*. We realized users are able to use key-typing game impartially without influence of the difference of communication latencies.

1 はじめに

ギガビットイーサネットなどの高速ネットワークが普及しつつある．また，高速ネットワークの普及により，今後ネットワークを介した即応性が要求されるアプリケーションの利用がさらに広まることが予想される．KDDとカプコンでは，70[msec]以内で届く高速性の対戦型ゲーム用ネットワークサービス「マッチングサービス」[1]の提供開始を2000年3月からはじめている．また，PlayStation2ではCATVと接続した高速なネットワークゲーム環境が計画されている．一方で，携帯電話・PHSを利用した比較的低速な無線回線によるモバイルコンピューティングが一般化してきている．このような高速ネットワークと低速ネットワークの混在する環境においては，即応性の要求されるネットワークアプリケーション，例えばネットワーク対戦ゲーム，オークション，電子会議システムなどを利用する際に，端末間の遅延差によりイベントパケットの順序がサーバ側で逆転する現象が起きる．それによりユーザが反応時間に対する不公平さ**図1**を感じたり，また因果関係が保たれなくなることがある．この結果，ネットワーク対戦ゲームで，回線のスピードが遅いために敵の動きを確認できなかったり，自分の動きが遅くなったりしてゲームに負けてしまうなどの問題が生じる．

筆者らは，上記のような不公平性の解消のための機構として，対戦ゲームにおけるメンバ間公平性保証方式（*ICEGEM*: Impartial Communication Environment for GamE Members）[2][3]を提案して

*静岡大学大学院理工学研究科
Graduate School of Science and Engineering, Shizuoka University
†愛知県立大学情報科学部
Faculty of Information Science and Technology, Aichi Prefectural University
‡静岡大学情報学部
Faculty of Information, Shizuoka University

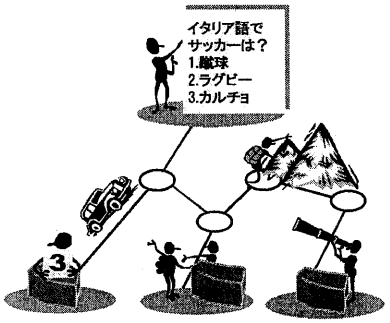

図 1 不公平性

いる．本方式は，反応時間に基づく公平性保証，クライアント側の制限時間，サーバ側の動的な制限時間・制限時間の短縮によりメンバ間で公平なサービスを受けることができる．

本稿では，本方式の詳細とプロトタイプによる評価について述べる．以下，まず端末間の遅延差による不公平性について述べ，関連研究とその問題を挙げ，本システムの反応時間に基づく公平性保証，クライアント側の制限時間，サーバ側の動的な制限時間・制限時間の短縮の詳細について述べる．次に，反応時間に基づく公平性保証を用いた，ネットワーク対戦型早押しキーボードタイピングゲームのプロトタイプについて述べ，遅延差が生じる端末間での評価実験について報告する．

2 端末間の遅延差による不公平性

2.1 逆転現象

高速なネットワークと低速なネットワークの混在する環境において，即応性の要求されるネットワークアプリケーションを利用する際に，端末間の遅延差によりイベントパケットの順序がサーバ側で逆転する現象が起きる．この逆転現象の例を**図 2** に示す．S はサーバ，C_1, C_2 はクライアント，R_1, R_2 はそれぞれのクライアントが表示されてから反応するまでの時間，B_1，B_2 はサーバがそれぞれのクライアントからのパケットを受信した時刻である．低速なネットワークを使用している C_1 は，高速なネットワークを使用している C_2 よりも早い時刻にパケットを送信したにもかかわらず，サーバ側では S-C_1，S-C_2 間の遅延差のために到着順序が逆転していることが分かる．このように，クライアント側でのイベントパケット送信時刻とサーバ側でのイベントパケットの到着時刻が逆転してしまうことにより，対戦ゲームなどにおいてクライアント間の不公平が生じる．

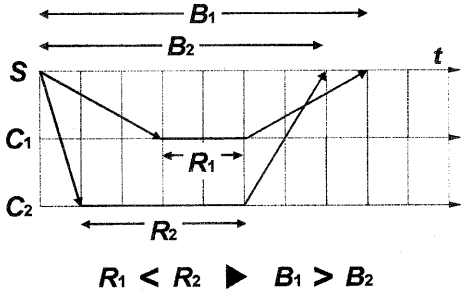

図 2 逆転現象

2.2 関連研究

このような逆転現象に対して因果関係の保存と，ユーザ反応時間との公平性を保証するための手法として [4][5][6][7][8] の研究がある．これらの研究は，サーバ側で一定時間，クライアントからのイベントパケットを待ち合わせた後，順序の制御を行っている．この一定時間の待ち合わせと，順序の制御を繰り返すことによって端末間の遅延差よる不公平性を解消している．ここで，サーバ側での一定の制限時間を設けている理由は，応答のないクライアントをいつまでも待っていては，即応性が損なわれるためである．そのため，たとえ制限時間を超えてからパケットが届いたとしてもサーバ側で破棄する手法が用いられている．

また，ネットワーク通信ゲームに特化したソフトウェア開発企業である株式会社ドワンゴは，高速通信ゲームシステム「DWANGO」 [9] を商品化している．このシステムは，アクションゲームなどのようにデータ伝送に同時性を要求するジャンルのゲームに適した高速通信ゲームシステムで，毎秒 60 フレーム同期通信を可能にする．現在，Dreamcast 用ネットワークゲームのサーバシステムとして採用されている．

2.3 問題点

[4][6][7] の研究では，NTP (Network Time Protocol) により全サーバ・クライアント間で時刻の同期をはかり，全クライアントの遅延時間を測定し，クライアントでの出力時刻を同期させる手法がとられている．出力時刻を同期させる理由は，特定の端末が早く情報を得ることによるクライアント間の不公平さをなくすなどの理由がある．また，全サーバ・クライアント間で時刻の同期を行い，それぞれのクライアントとサーバ間のメッセージ伝達遅延に基づいて，すべてのクライアント間での表示時刻を合わせることによって，クライアントのイベントパケット送信時刻から順序の制御を行っている．しかしながら，遅延時間は常に変動しているのでクライアント側で到着時刻をすべてのクライアントで完全に合わせることは難しい．

また，実験環境が比較的高速なネットワークであ

ること，クライアント数が少数であることから，低速・高速のネットワークが混在したのインターネット上での利用を考えると課題が残る．また，制限時間までに全クライアントからの反応パケットがサーバに到着しても制限時間まで待ち合わせてから順序の制御を行うため，サーバ側での無駄な待ち時間が発生する．そこで，筆者らは関連研究とは別のアプローチとして，全サーバ・クライアントでの時刻同期を必要としない，反応時間に基づく公平性保証を提案している [2][3]．本方式は，即応性を向上させ，クライアント側でも制限時間を設けることにより，無駄なトラヒックを軽減する．次節では，本方式の詳細について述べる．

3 ICEGEM

3.1 概要

関連研究で想定している環境は，全クライアント・サーバが比較的高速なネットワークに接続されており，通信がLANなどの狭い範囲で行われていたこと，クライアント数が少ないものであった．しかしながら，これは非常に限られた環境であり，対戦ゲームなどはLANに限らず，インターネット，無線ネットワーク，モデム接続など，遅延差の大きい環境で利用されることが予想される．そこで，比較的即応性の要求されるネットワークゲームを対象としたメンバ間公平性保証方式 (*ICEGEM*) を提案する．想定環境は，以下の通りである．

アプリケーション：ゲーム

ネットワーク：LAN，インターネット

通信速度：ギガビット～32Kbps

クライアント数：0～20

TCP：接続，離脱処理命令用

UDP：ゲーム処理命令用

接続・離脱といったパケットロスが許されない命令はTCPにより，ゲーム内での順序の制御を行うパケットロスが許される命令はUDPにより転送する．UDP上で転送するメッセージでは，即応性を重視しつつ，ユーザ間の公平性を保証するものとする．*ICEGEM* の主な4つの特長を以下に示す [2]．

■反応時間に基づく公平性保証
クライアント側の画面に表示されてからユーザが次のイベント動作を行うまでの反応時間により公平性保証する

■クライアント側の制限時間
クライアント側でユーザの反応時間に制限時間を設け，サーバ側にパケットを送信しないことでトラヒックの軽減，無駄な負荷を下げる

■サーバ側の動的な制限時間
サーバ側での制限時間を全クライアントの遅延時間の変動から動的に変更する

■サーバ側の制限時間の短縮
制限時間に関係なく，順序の制御が行える状態になれば，その時点で制御を行い制限時間を短縮する

3.2 詳細

3.2.1 記号の説明

以下の説明では，サーバがイベントをクライアントにイベントパケットを送信し，クライアントがそれに対する反応を返す1サイクルを1ターンと呼ぶことにする．また，*ICEGEM* で定義する記号を以下に示す．

S：サーバ

C_i：クライアント $(i = 1, \cdots n)$

T_j：j ターン目の S 側の制限時間 $(j = 1, \cdots \infty)$

L_j：j ターン目の C 側の制限時間

$R_{i,j}$：ユーザの反応時間

$B_{i,j}$：返答パケットの受信時間

$D_{i,j}$：反応時間を含めない往復遅延時間

$P_{i,j}$：反応時間を含めない往復遅延時間

3.2.2 反応時間に基づく公平性保証

1つ目の特長である反応時間に基づく公平性保証について説明する．まず，サーバが全クライアントにイベントパケットの送信を行う．各クライアントはパケットを受信し，反応命令と反応時間 $R_{i,j}$ の情報を付加し，サーバに送信する．サーバは各クライアントからの T_j までに受信したパケットに対し，クライアント側の反応時間 $R_{i,j}$ に基づいて順序の制御を行う．これにより，全クライアント・サーバ間で時刻の同期をとる必要はなく，またすべてのクライアントで同時に表示させなくても公平性を保証できる [3]．

3.2.3 クライアント側の制限時間

無駄なパケットの送信により，サーバへの負荷を軽減するために，サーバでの制限時間に加えて，クライアント側でも制限時間を設ける．この制限時間を超えたパケットはサーバに送信することなくその場で破棄する（**図3**）．サーバ側の制限時間 T_j は，最大往復遅延時間とクライアント側の制限時間 L_j の和である．したがって，L_j を超えたパケット

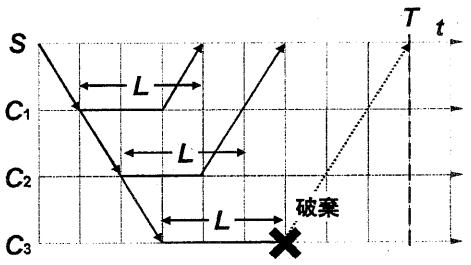

図 3 クライアント側の制限時間

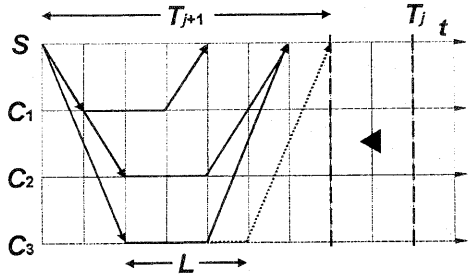

図 4 制限時間の変更

をサーバに送信してもサーバの制限時間を超える可能性が高いと考えられるためである．ここで，クライアント C_1 のように高速なネットワークに接続されているユーザはクライアント側の制限時間 L_j を超えたパケットを破棄せず，サーバに送信してもサーバ側の制限時間 T_j 内に到着可能であると予想できるが，公平性を保証するため，平等にパケットを破棄することにする．これにより，無駄なトラヒックの軽減をし，またサーバへの無駄な負荷を軽減する．

1 ターン内にパケットが到着しない場合として，(A) クライアント側の制限時間によるパケットの破棄，(B) 途中の通信路でのパケットロス，(C) サーバ側の制限時間によるパケットの破棄という 3 つケースが考えられる．プロトタイプシステムのようなゲームなど，パケット 1 つの重要性より即応性が重要となる場合には，(A), (B) のケースを区別する必要がないこともある．しかしながら，オークションのようなアプリケーションでは，パケットロスがアプリケーションのサービスを損なう恐れがある．本稿では，(A)，(B) を区別しないケースを想定する．また (C) のケースでは，受信されたパケットはアプリケーションのサービスに対して影響を与えないが，次のターン以降のクライアントの反応時間の予測に利用できる．

3.2.4 サーバ側の動的な制限時間

まず最初にサーバに対してクライアントが接続を行い，全クライアントが接続した段階で各クライアントの初期往復遅延時間 $D_{i,0}$ の設定を行うため，サーバは全クライアントに対してパケットを UDP により送信する．それに対し全クライアントは $R_{i,0} = 0$ でサーバに反応を返す．サーバは受信したパケットから各クライアントの初期往復遅延時間 $D_{i,0}$ の設定を行う．反応時間を含めない初期往復遅延時間は，$D_{i,0} = B_{i,0}$ となる．

1 ターン目のサーバ側の制限時間の初期値 T_1 は，予測最大往復遅延時間とクライアント側の制限時間 L_1 によって決定する．1 ターン目の予測往復遅延時間を P_1 とし，全クライアントでの最大の往復遅延時間を $\max(D_{i,1})$ とすると T_1 は，

$$P_1 = \max(D_{i,0})$$

$$T_1 = \beta P_1 + L_0$$

となる．なお，予測往復遅延時間 P_j については後で詳しく説明する．UDP によるパケットの欠落を考慮し，初期設定においてのみ往復遅延時間は全クライアントの情報が得られるまで繰り返し行う．

遅延時間の変動に伴い，1 ターンごとにサーバ側の制限時間の変更を行う．j ターン目のクライアントの往復遅延時間は受信した全クライアントからのパケットにより，$D_{i,j} = B_{i,j} - R_{i,j}$ となる．$(j+1)$ ターン目のサーバ側の制限時間 T_{j+1} は，$(j+1)$ ターン目の予測往復遅延時間 P_{j+1} とクライアント側の制限時間 L_{j+1} の和により，

$$P_{j+1} = \alpha P_j + (1-\alpha)\{\max(D_{i,1})\}$$

$$T_{j+1} = \beta P_{j+1} + L_{j+1}$$

と求められる．予測往復遅延時間は，前ターンの予測往復遅延時間と現ターンの最大の往復遅延時間の和とする．また，$\alpha(0 \leq \alpha \leq 1)$ は前ターンの予測遅延時間に与える重みづけで，小さいほど変動を大きくし，β は遅延変動を吸収するマージンである．α の値の調整で，ネットワークの状況に伴い常に変化する遅延時間に対応させ，次のターンでのサーバ側の制限時間 T_{j+1} を動的に変更する (**図 4**)．なお，現段階では T_j を超えて到着したパケットは，T_{j+1} を決定する際に $\max(D_{i,j})$ の計算に用いていないが，継続的にサーバ側の制限時間を超えるパケットが存在する場合には，これらのパケットの情報の利用が必要となる．

3.2.5 サーバ側の制限時間の短縮

図 5 のように，サーバ側の制限時間 T_j までに全クライアントからの返答パケットを受信した時点で順序の制御が開始できるので，その時点を 1 ターンとし，制限時間を短縮する．**図 6** に示すように，サーバは必ずしも制限時間 T_j までに全クライアントからの返答パケットを受信できるわけではない．これは，クライアント側の制限時間によって，パケットが送信されずに破棄されていたり，ネットワークの途中でパケットが欠落している場合がある

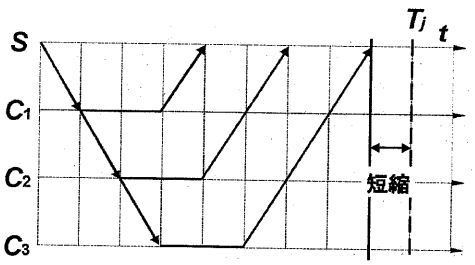

図 5 全パケット到着時の短縮

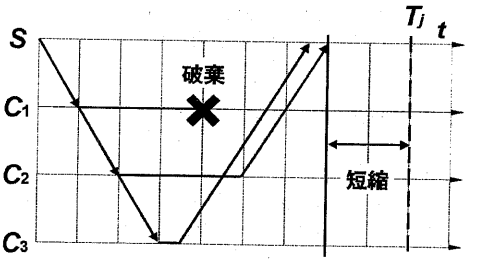

図 6 全パケット未到着時の短縮

からである．そこで本方式では，クライアントからの応答を得るたび，前ターンの最大の往復遅延時間を持つクライアントの往復遅延時間にクライアント側の制限時間を加算した時間まで待ち合わせる．一般に最大の往復遅延時間が，

$$\max_i(D_{i,j-1}) - L_j < D_{k,j-1}$$

を満たすクライアント k が存在し，そのうちサーバがまだ返答パケットを受信していないクライアントの集合を $\mathbf{Y}$ とすると，

$$T_j' = \beta \max_{k \in \mathbf{Y}}(D_{k,j-1}) + L_j$$

まで待ち合わせるか，もしくは，クライアント $k \in \mathbf{Y}$ からのパケットをすべて受信した時点で，サーバは順序の制御を開始する．この短縮により，サーバでの無駄な時間を省略することができ，即応性を向上させることができる．

4 実験

4.1 実験用タイピングゲーム

提案方式による公平性保証を評価するため，ネットワークを介した早押しキータイピングゲームを実装した [10]．このゲームは，多人数が異なるネットワーク環境からサーバに接続し，早押しキータイピングを行う対戦型のゲームである．サーバから 1Byte のゲームメッセージが全クライアントに同時に送信され，クライアントではメッセージを受信し，表示された 4 文字以内の単語を打ち込む．クライアントの画面に単語が表示された時刻から，正

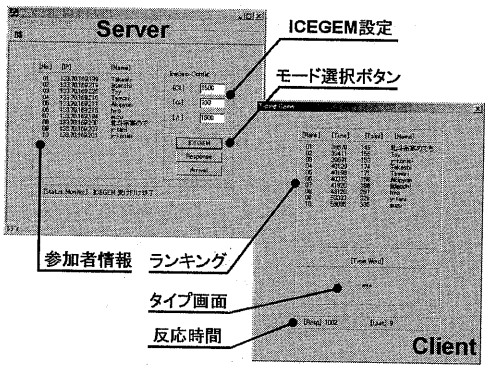

図 7 サーバとクライアントの画面

しくタイプされるまでの時間を反応時間とし，クライアントはサーバ側にその反応時間情報を送信する．サーバは全クライアントから送られてきた反応時間により，順位制御を行う．また，ターンごとに反応時間を足していき，総反応時間の短い順にプレイヤーのランキング結果を，クライアント側に送信する．最終的にゲーム終了時に総反応時間が最も短かったプレイヤーが優勝となるゲームである．

以上のゲームの命令メッセージを UDP により伝送する．一方，遅延時間の初期設定やサーバで集計された順位のフィードバック，その他接続確認などの命令メッセーを TCP により伝送する．**図 7** は左側がサーバプログラムにおいて，ICEGEM の設定を行う画面で，右側はクライアントのゲーム画面である．

4.2 実験環境

実験環境を**図 8** に示す．サーバと全クライアントは，同一の LAN に存在し 100Mbps で接続されており，擬似的な遅延をクライアント側で端末ごとにそれぞれ異なった値を設けている．Client 1 は，擬似遅延を 0 に設定した端末とし，Client 2 は，固定の遅延 1.0[sec] を設けた端末とし，Client 3 は固定の初期遅延 2.0[sec] から毎ターン 0.1[sec] 以内の誤差でランダムに増減する端末とした．また，突発的な大きな遅延変動として，5(%) の確率で片方向 3[sec] 程度の遅延が起こるように設定する．それぞれ平均往復遅延時間は，0.003[sec]，0.998[sec]，2.217[sec] となった．

4.3 実験内容

全ての実験で，1 ゲームのターン数を 50 とし，ランキング結果は TCP により 2 ターンに 1 回の割合でクライアント側に送信する。実験では，10 人のプレイヤーが異なる環境をランダムに交代しながら行うこととする．

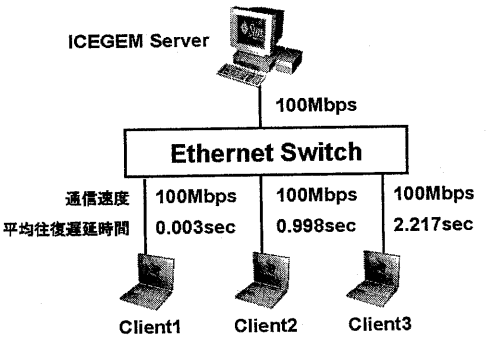

図 8 実験環境

まず，準備実験としてクライアント側の制限時間を適切な値に設定するための実験を行った．実験は8人のプレイヤーが交代しながら，α，βの値，クライアント側の制限時間を定めない状態で，合計600ターン4文字以内の単語をタイプしてもらった．結果からクライアント側の制限時間を0～2.0[sec]の間で設定した場合に得られるパケットの破棄率を**図9**に示す．0～0.5[sec]までは，すべてのプレイヤーは反応することができず，破棄されてしまう．また，1.0以下でも半数以上が制限時間に間に合わなくなり，ゲームが成り立たなくなる恐れがある．一方，2.0[sec]まで待ち合わせるとほとんどのパケットが破棄されない代わりに，ゲームスピードが遅くなってしまう．そこで，80(%)以上のパケットが破棄されることなく，プレイヤーがゲームに参加でき，かつゲームスピードを損なわない制限時間として，実験では1.5[sec]に設定することにする．

[Ex1] クライアント側の制限時間，サーバ側の動的な制限時間などの設定を行わずに，異なる3つのクライアント環境において，クライアントからのメッセージの到着順と反応時間による順序の比較を行った．**[Ex2]** さらに，多人数による利用ができるかを検証するため，クライアント数を3人，5人，10人と変化させた実験を行った．**[Ex3]** 同じ環境

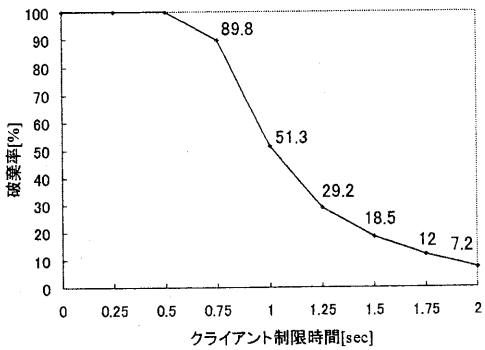

図 9 クライアントの制限時間

において，α，βをそれぞれ適切値[11]の0.75，1.5に設定し，クライアント側の制限時間を1.5[sec]として，サーバ側の動的な制限時間の変化を見る実験をした．**[Ex4]** 最後に，サーバ側の制限時間の短縮をした場合との比較を行うため，サーバ側の制限時間の短縮をなしにして実験した．それぞれの実験でのパラメータを以下に示す．

■ **Ex1 到着順と反応時間順**
クライアント数：3
αとβの値：定めない
クライアント側の制限時間：なし
サーバ側の制限時間の短縮：なし
ゲーム数：100ターン

■ **Ex2 多人数による利用**
クライアント数：3・5・10
α：0.75，β：1.5
クライアント側の制限時間：1.5[sec]
サーバ側の制限時間の短縮：あり
ゲーム数：100ターン

■ **Ex3 動的な制限時間**
クライアント数：3
α：0.75，β：1.5
クライアント側の制限時間：1.5[sec]
サーバ側の制限時間の短縮：あり
ゲーム数：100ターン

■ **Ex4 サーバ側の制限時間の短縮**
クライアント数：3
α：0.75，β：1.5
クライアント側の制限時間：1.5[sec]
サーバ側の制限時間の短縮：なし
ゲーム数：100ターン

5 評価

5.1 到着順と反応時間順の比較

サーバにクライアントからの反応時間メッセージパケットが到着する順序と，反応時間による順序の違いを調べるために，3つの異なる環境においての順位獲得数を比較することにする．**図10**は，左側が到着順による順位獲得数で，右側が反応時間順による順位獲得数である．図からも明らかなように，到着順ではクライアントの環境によって，順位獲得数に偏りが見られる．Client 1は遅延が最も小さかったため，200回中182回も1位を獲得していた．それに対し，Client 3は遅延が最も大きかったため，200回中184回も3位であった．しかしながら，反応時間順を見れば，実際はClient 3が最も1位を獲得していたのである．この実験のサーバ側の到着順序の逆転率は，94.5(%)であった．このように，遅延差のある環境においては到着順ではなく，

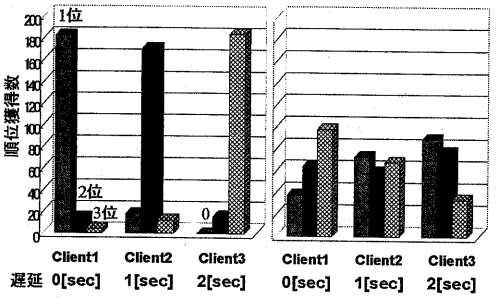

図 10 順位獲得数の比較

反応時間順で制御しないと不公平を生じてしまうのである.

5.2 多人数による利用

ICEGEM では複数のクライアントでの利用を想定しているので，今回は 3 人，5 人，10 人とプレイヤーを増やしていき，ゲームが安定して利用できるかを検証した．**図 7** は 10 人でプレイしている状態の画面であるが，3 人，5 人，10 人いずれも即応性を損なったり，パケットロスを起こすことなく全く問題なく利用できた.

5.3 動的な制限時間

ここでは，突発的な大きな遅延や緩やかに変動する遅延に対する，サーバ側の動的な制限時間の変化を見る．**図 11** に示されるように，Client 1，Client 2 の固定の遅延に対して，Client 3 は緩やかに変動する部分と，突発的な大きな遅延が現われる部分が見られる.

サーバ側の動的な制限時間は，この緩やかな変動を吸収して Client 3 のゲームへの参加を助ける役目を果たしていた．一方，突発的な大きな遅延に対しては，待ち合わせをしていては，ゲームのスピードを損ねる原因となるので，破棄することが求められる．そこで，サーバ側の動的な制限時間である

$$T_{j+1} = \beta P_{j+1} + L_{j+1}$$

の式を，**図 11** に示す．クライアント側の制限時間を引いた βP_{j+1} と $Client3$ の往復遅延時間を比較すると，突発的な大きな遅延のみを破棄していることが分かる.

しかし，実験で定めた β の値が大きかったせいで，サーバ側で待ち合わせる時間を余分に取りすぎたことが，即応性を損ねる原因となった．今後は遅延変動の大きさも考慮して，α，β の適切値を求める必要がある．実験から α，β の値は**表 1** のような作用をもたらすことが確認できた.

この実験では，クライアント側の制限時間により破棄されたパケットは全体の 22(%) であった．また，サーバ側の制限時間を超えたパケットは 11 回

表 1 α，β の値の影響

	遅延変動	ターン間隔 (ゲーム)
$\alpha \to 0$	敏感反応	不規則
$\alpha \to 1$	鈍感反応	滑らか
$\beta \to 1$	吸収小	速い
$\beta \to 2$	吸収大	遅い

であった．クライアント側で破棄された場合，サーバは遅延時間を前ターンの値で計算している．クライアント側の制限時間によりパケットが破棄されて，遅延情報が何ターンにも渡って欠落した場合，遅延時間が大きくなっていてもサーバ側は予測できなくなり，サーバ側の制限時間を継続して超えてしまう可能性がある．そこで，クライアント側の制限時間で破棄しないで，制限時間に間に合わなかったという情報をサーバに返答する手法を今後検討する必要がある.

5.4 サーバ側の制限時間の短縮効果

実験では，100 ターン中 88 ターンサーバ側で制限時間の短縮が行われた．約 90(%) のターンで短縮が行われていたことになる．時間にするとゲーム時間の合計 501.4[sec] の内の 151.6[sec]，全体の約 30(%) の時間を短縮させた．1 ターンでは平均 1.5[sec] の短縮が行われたことになる．このことから，サーバ側の制限時間の短縮により，ゲームの即応性を向上させる効果があった．しかしながら，この短縮はもともと β によってサーバ側の制限時間を余分に取っていた時間が短縮されたことになる．よって，β の値をもう少し小さくすれば，1 ターンの制限時間が減少し，短縮される時間も減少すると考えられる．また，あまり β の値を大きくしすぎると，短縮されるターンと短縮されないターンの時間差が激しくなり，プレイヤーがストレスを感じやすくなる原因となる．短縮による違和感を抑えるためにも β をなるべく小さくする必要がある.

6 おわりに

本稿では，メンバ間で公平なサービスを受けられるように，反応時間に基づく公平性保証，クライアント側の制限時間，サーバ側の動的な制限時間・制限時間の短縮を用いた対戦ゲームにおけるメンバ間公平性保証方式 (*ICEGEM*) を提案した．さらに，ネットワークを介した早押しタイピングゲームを実装し，実験により (*ICEGEM*) による効果を評価した．その結果，遅延差のあるネットワーク環境において，反応時間に基づく公平性保証を行ったものは，公平にゲーム上で順位を獲得することができた．また，サーバ側の動的な制限時間により，突発的な大きな遅延に対しては待ち合わせをせず，緩や

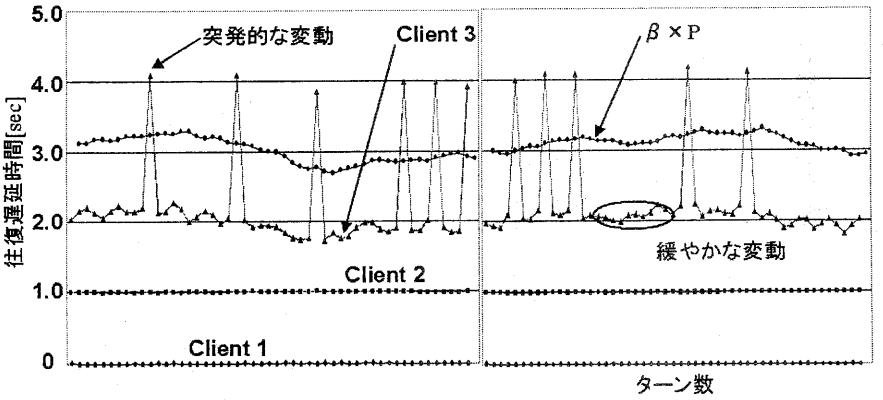

図 11 遅延変動と動的な制限時間

かな遅延変動に対しては遅延を吸収することでゲームへの参加を保証することができた．さらに，サーバ側の制限時間の短縮により，即応性を向上させることができた．

現状の問題点は，クライアント側の制限時間を超えたパケットをサーバ側に送信することなくクライアント側で破棄しているが，サーバ側の動的な制限時間の効果により，クライアント側の制限時間を超えたパケットがサーバで制限時間に間に合う現象が起きていることである．また，クライアント側で破棄することで，サーバが往復遅延時間の情報が受信できないという問題もある．このようなことを考えると，クライアント側の制限時間を超えた場合，制限時間に間に合わなかったという情報をサーバに送信する手法の方が，効果的であると考えられる．

さらに，突発的な遅延でクライアント側に遅れて到着したゲームメッセージが，次のターンのメッセージと順序が入れ代わってしまったり，すぐに次のターンに進行してしまうなどの現象が起きてしまっている．この対策としては，サーバでの送信時刻情報をゲームメッセージと合わせて送ることにより，クライアント側で送信時刻からある一定の時間が経過していれば，突発的な遅延によって遅れて到着したと判断し，そのターンの処理を省くなどを考えている．

今後の課題として，クライアントの反応時間の偽証行為についての対策の実装と，ゲーム参加前に全クライアントの遅延変動をある程度測定してから適切な α，β を設定するような手法の実装と，β の値に対するサーバ側の制限時間の短縮効果についての評価，そしてICEGEMで適用できるゲームアプリケーションの検討について取り組む予定である．

参考文献

[1] マッチングサービスの提供開始について：http://www.kdd.co.jp/press00/00-002r.html

[2] 石川貴士，石原進，井手口哲夫，水野忠則：リアルタイム性の強いネットワークアプリケーションの公平性を考慮した通信方式の提案，1999年電気関係学会東海支部連合大会，578，pp. 289，(1999.9)．

[3] 石川貴士，石原進，井手口哲夫，水野忠則：リアルタイム性の強いネットワークアプリケーションの公平性を保証した通信方式の提案，第59回情処全大，3V-4，pp. 551–552，(1999.9)．

[4] 徳永博之，関野公彦，久保田創一，佐藤栄：リアルタイムグループウェアにおけるイベント順序制御の一考察，情処学DPS研報，82-30，pp. 171–176，(1997.4)．

[5] 伊関宏心，富永英義：マルチメディアシステムにおけるイベント順序制御手法の提案，電子情報通信学会総合大会，B-7-174，pp. 295，(1998)．

[6] 桑子純一，瀬崎薫：分散環境におけるメディア同期，信学技報，SSE98-100，IN98-81，pp. 67–72，(1998.9)．

[7] 橘芳郎，石橋豊，田坂修二：Δ因果公平性保証を用いたメディア同期方式の性能評価実験，信学技報，SSE98-203，IN98-175，pp. 25–30，(1999.3)．

[8] Y. Ishibashi and S. Tasaka : A group synchronization mechanism for live media in multicast communications, in *Conf. Rec. IEEE GLOBECOM'97*, pp. 746–752, (1997.11).

[9] DWANGO : http://www.dwango.co.jp/

[10] 石川貴士，石原進，井手口哲夫，水野忠則：対戦ゲームにおけるメンバ間公平性保証方式：実装と評価，第6回モバイルマルチメディア通信ワークショップ，MoMuC-J6，pp. 1–8，(2000.3)．

[11] 石川貴士，石原進，井手口哲夫，水野忠則：対戦ゲームにおけるメンバ間公平性保証方式の性能評価，マルチメディア，分散，強調とモバイル(DICOMO2000)シンポジウム，Vo.2000，No.7，pp. 391–396，(2000.6)．