

## 遅延差のあるネットワークにおける メンバ間公平性保証方式の特性評価

石川 貴士<sup>†</sup>, 石原 進<sup>††</sup>  
井手口 哲夫<sup>†††</sup> 水野 忠 則<sup>††</sup>

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を解消するメンバ間公平性保証方式 (*ICEGEM*: Impartial Communication Environment for Game Members) を提案する。本方式では、ユーザの反応時間に基づいて各ユーザの操作に対する順序制御を行うことにより、サーバ・クライアントの時刻同期を必要としないで、端末間の遅延差がある場合でも時間に関して公平なユーザ操作環境を実現する。また、順序制御のための制限時間を設け、これを全クライアントの遅延の測定・予測に基づいて動的に変更させることによって、ネットワークのトラフィックの変動や端末の移動による通信環境の変化にともなう遅延変動に対応する。本稿では *ICEGEM* の詳細、および本方式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実装、遅延差が生じる端末間での実験について述べる。実験の結果、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、サーバ側の制限時間を動的に変更することにより、遅延変動があった場合でもゲームの即応性を保ったうえですべてのユーザが公平にゲームを行えることが確認できた。

### Performance Evaluation of a Method for Offering Impartiality to Game Members in Heterogeneous Network Environments

TAKASHI ISHIKAWA,<sup>†</sup> SUSUMU ISHIHARA,<sup>††</sup> TETSUO IDEGUCHI<sup>†††</sup>  
and TADANORI MIZUNO<sup>††</sup>

In real-time network applications, such as network games, the users often suffer event inversion on heterogeneous network environments including small delay networks and large delay networks. Because of this, the fairness between users can not be guaranteed. In this paper we propose a method for offering impartiality to game members who play on heterogeneous network environments *ICEGEM* (Impartial Communication Environment for Game Members). The order of the users' operations can be treated correctly at the server using users' operation time with *ICEGEM*. Another feature of *ICEGEM* is that it adjusts timeout values for synchronization of clients' messages dynamically so that it can be used in practical Internet environments and mobile computing environments. We implemented a network key-typing game with *ICEGEM*, and evaluated its effect. We confirmed that users are able to play the game impartially without the influence of the difference and fluctuations of delays.

#### 1. はじめに

ギガビットイーサネット・ADSL 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、今後ネットワーク対戦型アクションゲームのように、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。現在、サーバによる一局集中管理で 70 [msec] 以内の高速性の対戦型ゲーム用ネットワークサービス<sup>1)</sup>が行われている。一方で、携帯電話・PHS などの無線回線や、モデムを使用した低速回線も現在広く利用されている。このような高速・低速ネットワークの混在する環境や複数のルータを経由するインターネット環境においては、即応性の要求されるネットワークアプリケーション、たとえばネットワーク対戦ゲーム、

<sup>†</sup> 静岡大学大学院理工学研究科  
Graduate School of Science and Engineering, Shizuoka University

<sup>††</sup> 静岡大学情報学部  
Faculty of Information, Shizuoka University

<sup>†††</sup> 愛知県立大学情報科学部  
Faculty of Information Science and Technology, Aichi Prefectural University  
現在、有限会社カラビナシステムズ  
Presently with Carabiner Systems Inc.

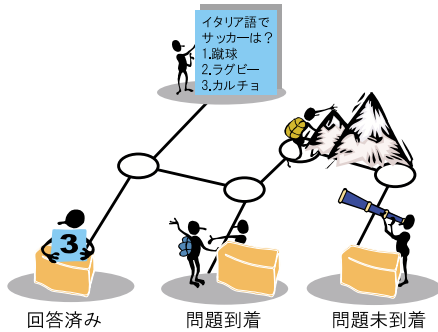


図 1 端末間の遅延差による不公平性

Fig. 1 Partiality caused by the difference of delay.

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メッセージの到着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して不公平さを感じたり（図 1），因果関係が保たれなくなったりすることがある．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．

また，インターネット環境ではトラフィックの変動によって遅延揺らぎが発生するし，移動無線環境では通信状態の変化やハンドオフなどによって遅延が大きく変動する．このように，各ホスト間の遅延は均一でないばかりかつねに変動する可能性がある．

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*）<sup>5)</sup>を提案している．本方式では，ユーザの反応時間に基づいて順序の制御を行うことにより，公平性を保証する．また，順序制御のための制限時間を動的に変更させることにより，トラフィックの変化や端末の移動による通信環境の変化にともなう遅延変動に対応することを目的とする．

本稿では，本方式の詳細とネットワークを介したタイピングゲームによる実験・評価について述べる．以下，2章で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点を示す．3章で本システムの反応時間に基づく公平性保証，動的制限時間の詳細について述べ，4章で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タイピングゲームの実装と実験について述べ，5章に評価結果を示し，6章でまとめとする．

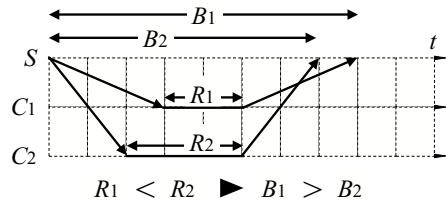


図 2 逆転現象

Fig. 2 Inversion phenomenon.

## 2. 端末間の遅延差による不公平性

### 2.1 逆転現象

高速・低速ネットワークの混在する環境や複数のルータを経由するインターネット環境においては，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．この例を図 2 に示す．図中の  $S$  はサーバ， $C_1, C_2$  はクライアント， $R_1, R_2$  はクライアント側で表示されてからユーザが反応するまでの時間， $B_1, B_2$  はサーバがイベントパケットを送信してから，クライアントからの応答パケットを受信するまでの時間である．遅延の大きい  $C_1$  は，遅延の小さい  $C_2$  よりも短い反応時間でパケットを送信したにもかかわらず，サーバ側では  $C_1, C_2$  と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．このように逆転現象により，対戦ゲームなどにおいてユーザ間に不公平が生じる．

### 2.2 関連研究

このような逆転現象に対して因果関係を保存する研究<sup>3),4)</sup>と，ユーザの反応時間の公平性を保証する<sup>5),6)</sup>研究がある．これらの研究では，サーバ側で一定時間，クライアントからの応答パケットを待ち合わせた後，順序の制御を行う方法を基本としている．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に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，サーバが応答のないクライアントをいつまでも待っているのは，ユーザの操作に対する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，ネットワーク対戦ゲームに特化したソフトウェア開発会社であるドワンゴは，高速通信ゲームシステム「*DWANGO*」<sup>7)</sup>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，DWANGOは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

文献3)~6)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行ったうえで，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．そして，クライアントの応答パケット送信時刻によって順序の制御を行っている．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ユーザ間の不公平さをなくすためである．しかしながら，遅延時間はネットワーク上のトラフィックの増減や端末の移動によるネットワークの環境の変化にともなって変動するため，全クライアントで表示時刻を完全に同期させることは困難である．

また，これらの文献で示されている実験環境は，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環境や複数のルータを経由するインターネット環境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応答パケットがサーバに到着しても制限時間が固定であるので，無駄な待ち時間が発生する．そこで，筆者らはこれらの関連研究の問題を解消し，全サーバ・クライアント間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方式を提案している．次章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，サーバと少数のクライアント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などの狭い範囲に制限されていた．しかし，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，即応性の要求されるネットワーク対戦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，以下のとおりである．

アプリケーション：ゲーム

ネットワーク：インターネット

クライアント数：数十台

TCP：接続，離脱処理用

UDP：ゲームメッセージ用

各メッセージのうち，サーバへの接続・離脱といった

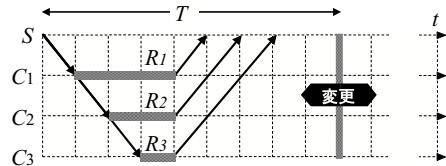


図3 ICEGEMの主な特長

Fig. 3 Main features of ICEGEM.

パケットロスが許されない処理をTCPにより伝送する．一方，ゲーム上で順序の制御を行う必要があるがパケットロスを許容できるゲームメッセージをUDPにより伝送する．UDPで伝送するメッセージは，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が適用できるゲームは，サーバが全クライアントに対して示す同じ条件に対して，全クライアントが応答するというシーケンスの繰返しにより行われるゲームである．この方式によるサーバ・クライアント間でのメッセージ交換を行う限り，単純な早押しクイズゲームから，スポーツゲームのように時々刻々と複数プレイヤー間で共有する空間の情報に変化するようなアクションゲームにまで適用可能である．

ICEGEMでは主に以下の2つの特長がある(図3)．

#### 反応時間に基づく公平性保証

クライアントで表示されてから，ユーザが反応するまでの反応時間に基づいて順序を制御することにより，時刻同期・端末間同期なしに公平性を保証する．

#### 動的制限時間

全クライアントの遅延時間の測定・予測に基づいて，メッセージ順序の判定のための制限時間を動的に変更させ，さらに無駄な待ち時間を短縮させ即応性を向上する．

### 3.2 記号の説明

以下の説明では，サーバが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応答パケットを返す1サイクルのことを1ターンと呼ぶことにする．本稿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( $i = 1, 2, \dots, n$ )

$T_j$ ： $j$ ターン目の $S$ 側の制限時間( $j = 1, 2, \dots, \infty$ )

$L_j$ ： $j$ ターン目の $C$ 側の制限時間

$R_{i,j}$ ：ユーザの反応時間

$B_{i,j}$ ：反応時間を含むパケット往復時間

$D_{i,j}$ ：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予測往復遅延時間

### 3.3 反応時間に基づく公平性保証

サーバは全クライアントに対してイベントバケットの送信を行う。各クライアントはバケットを受信すると、そのイベントに対するユーザの応答後に、応答メッセージとユーザの反応時間  $R_{i,j}$  の情報を付加し、サーバへ送信する。サーバは各クライアントからの  $T_j$  までに受信したバケットに対し、反応時間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がなく、またすべてのクライアントで表示時刻を同期させることなく公平性を保証できる。

### 3.4 動的制限時間

#### 3.4.1 クライアント側の制限時間

ICEGEMでは、毎ターンのサーバ・クライアント間の往復遅延時間の測定を行う。往復遅延時間の測定は、バケット往復時間  $B_{i,j}$  からクライアントで測定するユーザの反応時間  $R_{i,j}$  を引くことにより行う。クライアント側では、ユーザの反応時間の測定と同時に、ユーザの反応が十分に速いかどうかの判定を行う。この判定のためのしきい値をクライアント側の制限時間と呼ぶことにする。反応時間がクライアント側の制限時間を超過した場合、ユーザの応答メッセージは、タイムオーバー情報とともに自動的にサーバに送信される。クライアント側に制限時間を設けることによって、遅延の小さいクライアントが遅延の大きいクライアントに比べて、反応に長い時間を要してもサーバ側の制限時間に間に合うような不公平性を解消することができる。さらに、ゲームに応じて適切なクライアント側の制限時間を設定することができるという利点もある。

ここで、1ターン内にクライアントからサーバへ応答メッセージが到着しない場合として、(i) 通信路の途中でのパケットロス、(ii) 突発的な遅延増加にともない、サーバ側の制限時間でパケット破棄という、2つのケースが考えられる。ICEGEMで想定しているゲームなどのアプリケーションでは、パケット1つの重要性より即応性が重要となるので、(i)、(ii)のケースを区別する必要がないこともある。一方、ネットワークオークションのようなアプリケーションでは、即応性よりも信頼性を重視する必要があるので、(i)のケースに対しては、再送処理を、(ii)のケースに対しては、制限時間を長めに設定するような対策が考えられる。

#### 3.4.2 サーバ側の動的な制限時間

サーバ側では、クライアントからの応答メッセージを受信するための制限時間を動的に変更する。サーバ側の制限時間  $T_j$  を、全クライアントの遅延時間の測定から最大の遅延時間を持つクライアントの最大往復

遅延時間  $D_{i,j}$  に、クライアント側の制限時間  $L_j$  を加算した値に設定すれば、全クライアントの応答メッセージを受信できる可能性が高くなる。しかしながら、遅延時間はつねに変動している。そこで、ICEGEMでは各クライアントの遅延時間の予測を行い、これらの最大値に基づいて制限時間  $T_j$  を変動させる。

クライアント  $i$  の予測往復遅延時間を  $P_{i,j}$  とすると、往復遅延時間の予測にTCPで用いられている往復遅延時間の予測式<sup>9)</sup>を用いると、サーバ側の制限時間は以下のように計算できる。

$$T_j = \beta \max_{i=1,2,\dots,n} (P_{i,j+1}) + L_j \quad (1)$$

$$P_{i,j} = \alpha P_{i,j-1} + (1 - \alpha) D_{i,j-1} \quad (j \geq 2) \quad (2)$$

$j$  ターン目のサーバ側の制限時間  $T_j$  は、 $j$  ターン目の予測往復遅延時間  $P_{i,j}$  の最大値とクライアント側の制限時間  $L_j$  の和となる。また、クライアント  $i$  の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往復遅延時間の和となる。ここで、 $\alpha$  ( $0 \leq \alpha \leq 1$ ) は前ターンの予測往復遅延時間に与える重みづけで、大きいほど変動を小さくする。 $\beta$  は遅延変動を吸収するパラメータで、大きいほど制限時間を長くする。 $\alpha$  の値の調整で、ネットワークの状況により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。なお、現在の実装では  $T_j$  を超えて到着したパケットは、 $T_{j+1}$  を決定する際に  $D_{i,j}$  の計算に用いていないが、サーバ側の制限時間を超えるパケットが継続して起こる場合には、これらのパケットの情報の利用が必要となる。

サーバ側の初期制限時間  $T_1$  は、次のように設定する。

$$T_1 = \beta \max_{i=1,2,\dots,n} (P_{i,1}) + L_1 \quad (3)$$

$$P_{i,1} = D_{i,0} \quad (4)$$

ここで、 $D_{i,0}$  はセッション開始時にあらかじめ測定したクライアント  $i$  とサーバ間の往復遅延時間である。また、初期測定においてのみ遅延情報が得られるまで繰り返し行う。

#### 3.4.3 サーバ側の制限時間の短縮

サーバ側の制限時間  $T_j$  までに全クライアントからの応答バケットを受信した場合、順序の制御が可能になるので、その時点をも1ターンとし、制限時間を短縮する。図4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応答バケットを受信できるわけではない。これは、通信路の途中でパケットロスを起こしていたり、突発的な遅延増加により応答バケットの到着が遅れている場合があるからで

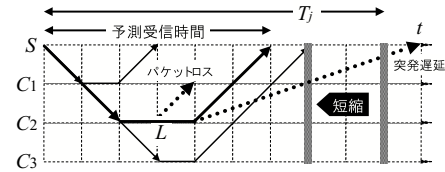


図 4 一部のクライアントからのパケット未到着時の短縮  
Fig. 4 Shortening of timeout value after a packet loss.

ある．そこで本方式では，遅延時間の予測から予測受信時間を計算し，予測受信時間を超過したクライアントからの応答パケットは，パケットロスかあるいは突発的な遅延増加が起きているものと判断し，短縮処理を行う．予測受信時間は，クライアントの往復遅延時間と  $\beta$  との積に，クライアント側の制限時間  $L_j$  を加算した時間とする．すなわち， $j$  ターン目における未受信クライアント  $k$  の集合を  $Y$  とすると，

$$\hat{T}_j = \beta \max_{k \in Y} (P_{k,j}) + L_j \quad (5)$$

まで待ち合わせる．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

提案方式の公平性・遅延予測・即応性の向上について評価するため，多人数参加型ネットワーク対戦早押しタイピングゲームを実装した<sup>8)</sup>．サーバから 1 パケット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，クライアントではゲームメッセージを受信し，表示された 4 文字以内の単語をユーザが打ち込む．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，順位の判定を行う．また，ターンごとに反応時間を加算していき，総反応時間の短い順にプレイヤーのランキング結果を作成し，クライアント側に転送する．最終的にゲーム終了時，総反応時間が最も短かったプレイヤーが優勝となる．

ゲームメッセージは UDP により伝送する．一方，遅延時間の初期測定やサーバで作成されたランキング結果，その他接続確認などのメッセージは TCP により伝送する．図 5 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．

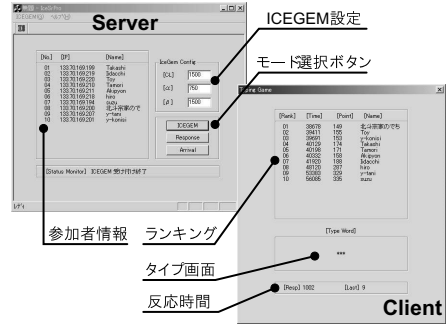


図 5 実験用ゲームのサーバとクライアントの画面  
Fig. 5 Screen-shot of server and client software of an experimental network game.

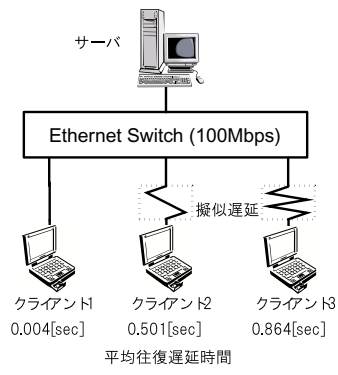


図 6 実験環境  
Fig. 6 Experiment environment.

### 4.2 実験環境

実験環境を図 6 に示す．サーバと全クライアントは，伝送速度 100 Mbps の Ethernet LAN に接続した．インターネット上での遅延差をエミュレートするために，クライアント内で端末ごとにそれぞれ異なる擬似遅延を設けた．クライアント 1, 2 には，それぞれ 0 [sec], 0.25 [sec] の固定の片方向遅延をクライアント内で与えることにした．クライアント 3 では初期片方向遅延 0.5 [sec] から毎回  $\pm 0.05$  [sec] 以内の範囲でランダムに増減させた．また，突発的な遅延増加として，5(%) の確率で片方向 2 [sec] 程度の遅延が起こるように設定した．各クライアントでの平均往復遅延時間は，それぞれ 0.004 [sec], 0.501 [sec], 0.864 [sec] となった．

### 4.3 実験内容

まず，実験に先だってタイピングゲームに適したクライアント側の制限時間を設定するために，プレイヤーの反応時間を調査した．実験は 8 人のプレイヤーに，4 文字以内の単語を合計 1000 ターン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]

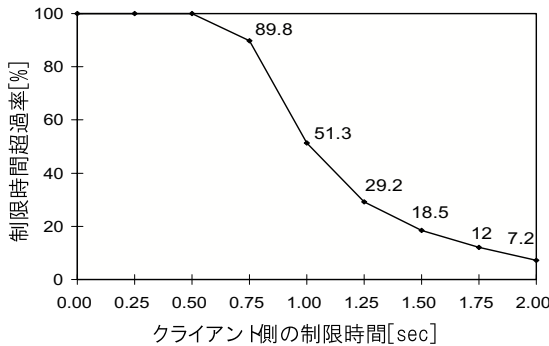


図 7 クライアント側の制限時間の設定  
Fig. 7 Setting of client-side timeout.

の間で設定した場合に得られるパケットの制限時間超過率を図 7 に示す．1.0 [sec] 以下では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] 以上待ち合わせるとほとんど間に合うようになる代わりに，ゲームスピードが遅くなってしまふ．そこで，プレイヤーがある程度ゲームに参加でき，かつゲームスピードを損なわない制限時間  $L_j$  ( $j = 1, 2, \dots, \infty$ ) として，実験では 80(%) 以上が制限時間に間に合う 1.5 [sec] に設定することにした．また，式 (1)，(2) における遅延予測に使用するパラメータ  $\alpha$  と  $\beta$  には，それぞれ予備実験より得られた適切値<sup>2)</sup> 0.75，1.5 を用いた．

以下に実験の内容を示す．

実験 1: 3 人のプレイヤーがそれぞれクライアント 1, 2, 3 において 100 ターンのゲームを行った場合の，応答パケットのサーバ到着順およびユーザの反応時間に基づく順位を獲得ポイントを選定した．

実験 2: 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, 5, 10 と変化させ，100 ターンの総ゲーム時間を測定した．

実験 3: 3 人のプレイヤーがそれぞれクライアント 1, 2, 3 を利用した場合の遅延時間とサーバ側の制限時間の変化を測定した．

実験 4: 実験 3 と同様のパラメータを設定し，3 人のプレイヤーのクライアント環境をすべてクライアント 3 として遅延変動とサーバ側の動的な制限時間を測定した．

実験 5: ping コマンドで静岡大学から yahoo.co.jp までの往復遅延時間を測定し，実験 3 と同様の値を設定した場合の，サーバ側の制限時間の変化を測定した．

実験 6: クライアントを PHS (64k) から静岡大学内の PIAFS サーバを介して yahoo.co.jp に接続

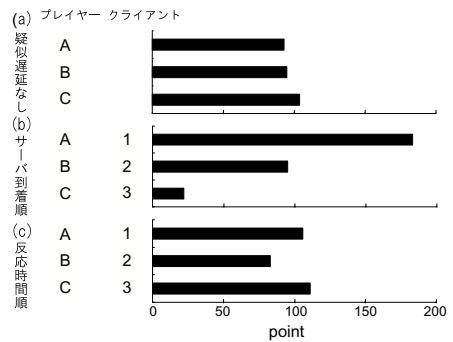


図 8 実験 1: 順位獲得ポイントの比較  
Fig. 8 Comparison of ranking points (Ex.1).

した場合の往復遅延時間を，ping コマンドにより測定し，実験 5 と同様にサーバ側の制限時間の変化を測定した．

実験 7: 3 人のプレイヤーのクライアント環境をすべてクライアント 3 として，サーバ側の制限時間の短縮時間を測定した．

## 5. 評価

### 5.1 到着順と反応時間順の比較

実験の結果に基づいて，サーバにクライアントからの応答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，クライアント 1, 2, 3 における順位獲得ポイントと比較することにする．順位獲得ポイントとは，1 位: 2 point, 2 位: 1 point, 3 位: 0 point とした場合の合計ポイントである．図 8 (a) は，各クライアント・サーバを同一の高速 LAN に接続し，疑似遅延を与えない公平な環境における各プレイヤーの成績を示したものである．(b) がサーバ到着順，(c) が反応時間順に基づく順位獲得ポイントである．図からも明らかなように，サーバ到着順の場合ではクライアントの環境によって，順位獲得ポイントに偏りが見られる．クライアント 1 は遅延が最も小さかったため，100 ターン中 89 ターンも 1 位を獲得していた．一方，クライアント 3 は遅延が最も大きかったため，100 ターン中ほとんどが 3 位であった．しかしながら，反応時間順の場合，疑似遅延を加えない場合とほぼ同様に平均的に順位を獲得している．図 8 (b) の実験での到着順序の逆転率は，66(%) であった．これらより，遅延差のある環境においては到着順による順序判定は不公平なものとなるが，反応時間順で制御することによりこの不公平を解消できることが確認できた．

### 5.2 多人数による利用

実験 2 の結果に基づいて，3 人，5 人，10 人とプレイヤー数を変化させたときの，総ゲーム時間の変化

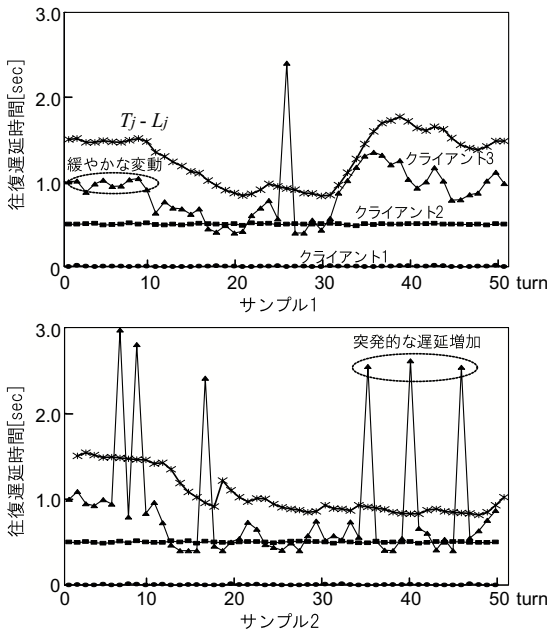


図 9 実験 3: クライアント 1, 2, 3 の場合の制限時間の変動  
Fig. 9 Fluctuations of server-side timeout (client 1, 2, 3: Ex.3).

を検証した．総ゲーム時間は，それぞれ 349.8 [sec] , 372.9 [sec] , 360.1 [sec] となり，プレイヤー数に関係ないことが分かった．図 5 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うことなくまったく問題なく利用できた．

5.3 サーバ側の動的な制限時間

ここでは，突発的な遅延増加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．実験 3 の結果を図 9 に示す．図中には式 (1) 中の  $T_j - L_j$  もあわせて示す． $T_j - L_j$  はサーバ側の制限時間  $T_j$  からクライアント側の制限時間  $L_j$  を差し引いた値である．

図 9 では，クライアント 1, 2 とサーバ間の遅延は大きさが小さく，変動も少ないが，クライアント 3 とサーバ間の遅延は，緩やかかつ大きく変動する部分と，突発的に増加する部分が見られる．緩やかな遅延変動に対しては，サーバ側での待ち合わせ時間の動的な変更は追従している．この結果，緩やかな変化の結果クライアント 3 との間の遅延が大きくなった場合においても，クライアント 3 の応答メッセージがサーバ側で破棄されていない．一方，突発的な遅延増加に対しては，待ち合わせをしては，ゲームのスピードを損ねる原因となるので，クライアント側の応答メッセージを破棄することが求められる．サーバ側の動的な制限時間  $T_j - L_j$  とクライアント 3 の往復遅延時間を

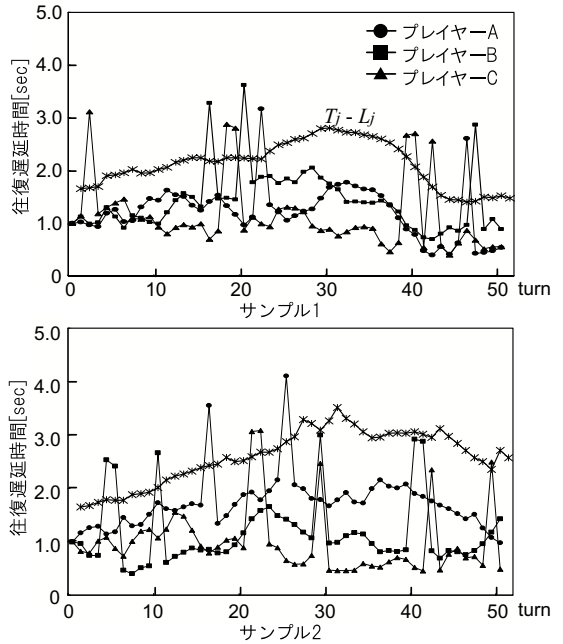


図 10 実験 4: クライアント 3x3 の場合の制限時間の変動  
Fig. 10 Fluctuations of server-side timeout (client 3 x 3: Ex.4).

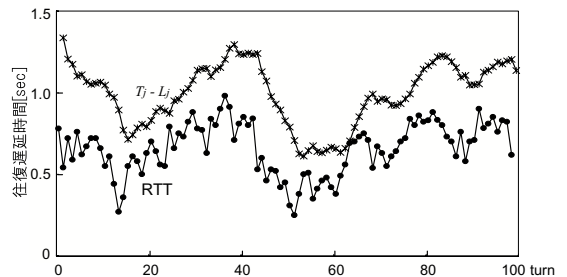


図 11 実験 5: 静岡大学-Yahoo 間の往復遅延と  $T_j - L_j$  の変化  
Fig. 11 RTT (Shizuoka Univ.-Yahoo) and  $T_j - L_j$  (Ex.5).

比較すると，突発的な遅延増加のみを破棄していることが分かる．

実験 4 で得られた遅延変動と  $T_j - L_j$  を図 10 に示す．実験 3 と同様にサンプル 1 のゲームでは，突発的な遅延増加のみをうまく破棄している．

実験 5 で得られた実際のインターネット接続時の往復遅延時間とその往復遅延時間に対して式 (1), (2) を適用した場合の  $T_j - L_j$  を図 11 に示す．緩やかな遅延変動に対して待ち合わせ時間が変化しており，遅延変動がうまく吸収されている．また，実験 6 の PHS 経由でのインターネット接続時の往復遅延時間と  $T_j - L_j$  を図 12 に示す．測定は PHS を屋内で静止させた状態で行った．実験 5 の場合と同様に，緩や

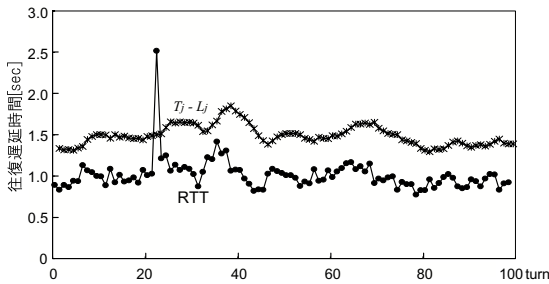


図 12 実験 6: PHS を介した場合の往復遅延と  $T_j - L_j$  の変化  
Fig. 12 RTT (PHS-Univ.-Yahoo) and  $T_j - L_j$  (Ex.6).

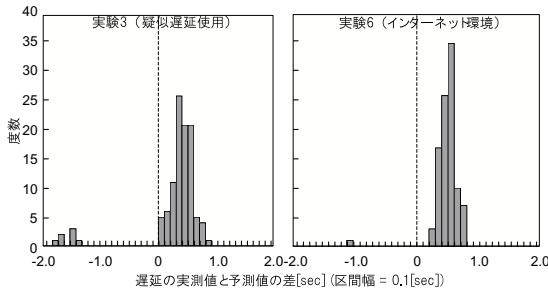


図 13 遅延時間の実測値と予測値の差のヒストグラム  
Fig. 13 Difference between RTT and  $T_j - L_j$ .

かな遅延変動は吸収し、突発的な遅延増加に対してはパケットを破棄していることが確認できた。

遅延予測の精度に関する定量的な評価として、疑似遅延を使用した実験 (実験 3) と実際のインターネット環境の実験 (実験 6) の両方における最大往復遅延時間の実測値  $\max_i(D_{i,j})$  と予測値  $T_j - L_j$  の差  $d_j = T_j - L_j - \max_i(D_{i,j})$  のヒストグラムを示す (図 13)。実験 3, 実験 6 のそれぞれの  $d_j$  の平均は 0.397 [sec], 0.497 [sec] であった。どちらの実験においても、突発的な遅延増加の場合を除くと、 $d_j$  はすべて 0~1.0 [sec] の間に存在し、負になることはなかった。すなわち、予測値を小さく見積もりすぎることによって、パケットロスがなく無事にサーバに到着しているクライアントの応答メッセージを破棄することはなかった。また、 $d_j$  が 1.0 [sec] より大きくなることはなく、遅延予測が即応性を損なわない程度であることも確認できた。この結果から、遅延時間の予測式 (1), (2) は妥当であると考えられる。

#### 5.4 サーバ側の制限時間の短縮効果

実験 7 では、100 ターン中 93 ターンサーバ側で制限時間の短縮が行われた。総ゲーム時間は 434.6 [sec] で総短縮時間は 80.9 [sec] となり、約 20 (%) 近くの時間を短縮させたことになる。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といえる。サーバ側の制限時間の短縮が行われた回数の内訳は、制限時間以内に全パケットが到着した場合が 79 ターンで、遅延時間の小さい端末の未受信パケットがある場合が 14 ターンであった。

## 6. おわりに

本稿では、端末間の遅延差が不均等かつ動的に変動するネットワーク環境でも、ネットワークゲームなどアプリケーションの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、動的制限時間を用いた公平性保証方式 (ICEGEM) を提案した。さらに、ICEGEM を用いて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、実験によりその効果を検証した。この結果、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、サーバ側の制限時間を動的に変更することによって、遅延変動があった場合でも、ゲームの即応性を保ったうえで、すべてのユーザが公平にゲームを行うことを可能であることが確かめられた。また、状況に応じたサーバ側の制限時間の短縮により、ユーザ操作に対するアプリケーションの即応性を向上させることができた。

現状の問題点は、ユーザの反応時間を偽証された場合への対処ができないことである。このような偽証行為への対策として、(i) 正しい認証コードを持つ信頼されたクライアントプログラムをサーバが配布し、認証されたクライアントからの正当なユーザの反応時間測定結果しか受け付けないようにする方法、(ii) 他のプログラムにより遅延を測定し、反応時間の偽りを判定する方法が考えられる。今回は遅延予測といった視点からアプリケーションをゲームに想定したが、本方式の 1 つの応用例は、オークションのようなアプリケーションへの適用である。ゲームではパケットロスや突発的な遅延増加に対して破棄を行っているが、オークションでは制限時間を大きく設定する必要があること、制限時間によってパケットを破棄可能であるかの検討が問題となる。即応性と信頼性のどちらかを重視するかについてのトレードオフが必要である。

## 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: 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) Ishibashi, Y. and Tasaka, S.: A group synchronization mechanism for live media in multicast communications, *Conf. Rec. IEEE GLOBECOM'97*, pp.746-752 (1997).
- 4) 橋 芳郎, 石橋 豊, 田坂修二: 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).
- 5) 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, *情報処理学会 DPS 研究会報告*, Vol.82, No.30, pp.171-176 (1997).
- 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).
- 7) DWANGO. <http://www.dwango.co.jp/>
- 8) 石川貴士, 石原 進, 井手口哲夫, 水野忠則: メンバ間公平性保証方式の同期機構の特性評価, *情報処理学会研究報告*, *モバイルコンピューティングとワイヤレス通信*, Vol.2000, No.15, pp.81-88 (2000).
- 9) Postel, J.: Transmission Control Protocol, RFC 793 (1981).

(平成 12 年 12 月 18 日受付)

(平成 13 年 5 月 10 日採録)



石川 貴士 (学生会員)

昭和 51 年生. 平成 11 年静岡大学工学部知能情報工学科卒業. 平成 13 年同大学大学院理工学研究科博士前期課程 (計算機工学専攻) 修了. 修士 (工学). 在学中は遠隔協調作業に関する研究に従事. 平成 13 年カラピナシステムズ (有) 入社. 情報システム向けソフトウェア製品の研究開発に従事.



石原 進 (正会員)

昭和 47 年生. 平成 6 年名古屋大学工学部電気学科卒業. 平成 11 年同大学大学院博士後期課程 (電子情報学) 修了. 平成 10 年度日本學術振興会特別研究員. 平成 11 年静岡大学情報学部情報科学科助手. 博士 (工学). モバイルコンピューティング, 無線環境用 TCP/IP に関する研究に従事. 電子情報通信学会, ACM 各会員. 平成 9 年度電気通信普及財団賞.



井手口哲夫 (正会員)

昭和 24 年生. 昭和 47 年電気通信大学通信工学科卒業. 同年三菱電機 (株) 入社. 平成 10 年愛知県立大学情報科学部地域情報科学科教授. 工学博士. ネットワークアーキテクチャ, LAN, 通信プロトコル設計方式, モバイルコンピューティング, タイムクリティカル通信等の研究に従事. 著書としては「コンピュータネットワーク概論」(ピアソン・エデュケーション), 「分散システム入門」(近代科学社), 「分散オペレーティングシステム」(科学技術出版, 訳書) 等多数. 電子情報通信学会, IEEE 各会員.



水野 忠則 (正会員)

昭和 20 年生. 昭和 43 年名古屋工業大学経営工学科卒業. 同年三菱電機 (株) 入社. 平成 5 年静岡大学工学部情報知識工学科教授, 現在, 情報学部情報科学科教授. 工学博士. 情報ネットワーク, プロトコル工学, モバイルコンピューティングに関する研究に従事. 著書としては「プロトコル言語」(カットシステム), 「コンピュータネットワーク概論」(ピアソン・エデュケーション) 等がある. 電子情報通信学会, IEEE, ACM 各会員. 当会フェロー.