

## ネットワーク環境に基づく P2P ネットワークゲームのグループ化

十川 基<sup>†</sup> 齊藤 裕樹<sup>‡</sup> 戸辺 義人<sup>‡</sup><sup>†</sup>東京電機大学大学院 工学研究科 情報メディア学専攻<sup>‡</sup>東京電機大学 工学部 情報メディア学科

## 1 はじめに

近年、コンピュータ相手ではなくプレーヤ同士がネットワークを通じてゲームを行うマルチプレーヤオンラインゲーム(MOG)が普及している。MOGは一般にクライアント/サーバ型のネットワークを利用する為、電源、保管場所、メンテナンス等コストが発生する。これらの欠点を補うために MOG を P2P 型のネットワークで実現する方法が提案されている。しかし公平性の維持やデータをどのように分散処理するか、スケーラビリティの維持、トポロジの形成方法等の課題がある。そこで我々は遅延をベースにしたプレーヤのポリシーに基づいて P2P のトポロジを形成するためにノード間の相対的な距離を示す VE map (Virtual Environment map)を提案する。またこの VE map を用いて効率的に P2P トポロジを形成する手法 EDMA(Efficient Delay Mapping Algorithm)を提案、評価を行う。

## 2 課題および前提条件

従来のクライアント/サーバ型のネットワークゲームでは地理やネットワークの条件により各クライアントとサーバまでの遅延時間に差がある。遅延の大きなノードに同期させると、ゲームのリアルタイム性が損なわれる為、データの到着順に処理され、サーバから遅延の少ないクライアントと遅延の大きなクライアントと公平性が均一に保つことができない。しかし P2P で処理を行うことにより、クライアントとしてのみ使用していた PC がサーバの役割を担うことも可能となる。そこで本研究ではユーザが遅延の少ない(i.e. 地理的に近い、同一ネットワークセグメント,同一 AS 等)ネットワークのユーザを効率的に見つけ、P2P トポロジを生成しグループ化する手法を提案する。対象とする環境はネットワークで接続された複数プレーヤ間で仮想空間を共有するゲームを想定する。ノードはネットワークの遅延を反映する地図として VE map を作成する。そのとき、各ノードの座標決定方法と、参加するグループを決定するアルゴリズムを定める。VE map では座標をノード間の遅延を基に決め、VE map 上の座標はノードとの相対的な遅延時間を表し、ノード間の距離が小さければ遅延が小さく、大きければ遅延が大きいことを示す。アルゴリズムはプレーヤのリストを保持するロビーサーバ  $S$  が存在し、ロビーサーバ  $S$  のアドレスはゲームのプロバイダから事前に告知され、すべてのノードが既知である。ロビーサーバ  $S$  は VE map 上の座標とそのアドレス、ノードがどのグループに所属しているかのグループ情報を提供する。各プレーヤ  $i$  ( $i=1, \dots, n$ )は自グループ番号( $g_i$ )と同じグループのノードの座標( $x_i, y_i$ )とアドレスのリストを所持する。ノード  $i$  のノード間遅延の許容最大値  $D_i$  とし、半径( $r_i$ )として VE map に反映させる。この閾値よりも値が小さい

グループをグループの候補( $G_{candidate}$ )とする。ノード  $i, j$  間の遅延は  $d_{ij}$  となり、VE map 上では  $r_{ij}$  と対応付けられる。ノード  $i$  からグループ内で一番遠いノードまでの距離を  $l_{max}$  とする。グループの中で一意に保たなければならないデータ管理するグループの代表ノードとしてスーパーノード  $SN$  とする。

## 3 提案手法

## 3.1 基本設計

本提案の基本的な動作概要を下記に示す。

1. 参加するノードはロビーサーバ  $S$  から VE map と各ノード情報(座標, グループ, アドレス)を得る。
2. VE map 上の概略の位置を決める REP(Rough Estimation Phase)を実行する。
3. 正確に座標を再測定する PEP(Precise Estimation Phase)を実行する。
4. いずれのグループとも条件が合わなかった場合は新しいグループを作成し、終了する。

## 3.2 REP

Algorithm 1 Rough Estimation Phase Algorithm.

```

1: procedure ROUGH
2:   if  $n < 3$  then
3:     measure delay with all current nodes
4:   else //  $n \geq 3$ 
5:     measure delay with 3 random nodes
6:   end if
7:   calculate rough ( $x_i, y_i$ )
8: end procedure

```

## 図 1. REP のアルゴリズム

図 1 に REP のアルゴリズムを示す。ノード  $i$  (プレーヤ  $i$ ) が MOG に参加するときは、ロビーサーバ  $S$  から VE map, 各ノードの情報を取得する。参加するノードは無作為に選び出した 3 つのノードから RTT を測定し、概略の位置( $x_i, y_i$ )を決める。ゲームに最初のノードが参加する場合、そのノードの座標は (0,0) となる。2 番目に参加するノードは (0,0) の円周上の座標となる。

## 3.3 PEP

Algorithm 2 Precise Estimation Phase Algorithm.

```

1: procedure PRECISE
2:   sort the groups in Gcandidate according to  $l_{max}$  in ascending order
3:    $g_{test} \leftarrow$  first group in Gcandidate
4:   repeat
5:     for all nodes in  $g_{test}$  do
6:       measure delay
7:     end for
8:     if the delays to all nodes in the group  $< D_i$  then
9:       calculate exact ( $x_i, y_i$ )
10:       $g_i \leftarrow g_{test}$ 
11:      node  $i$  enters this group
12:     else
13:        $g_{test} \leftarrow$  next group in Gcandidate
14:     end if
15:   until (node  $i$  found an appropriate group) or (all groups in Gcandidate has been used up)
16: end procedure

```

## 図 2. PEP のアルゴリズム

Measurement-based Peer-to-Peer Grouping for Networked Games

<sup>†</sup>Hajime Sogawa

<sup>‡</sup>Hiroki Saito

<sup>‡</sup>Yoshito Tobe

Dep. of Info and Media Engineering Tokyo Denki Univ (<sup>†</sup>)

Dep. of Info Systems and Multimedia Design Tokyo Denki Univ (<sup>‡</sup>)

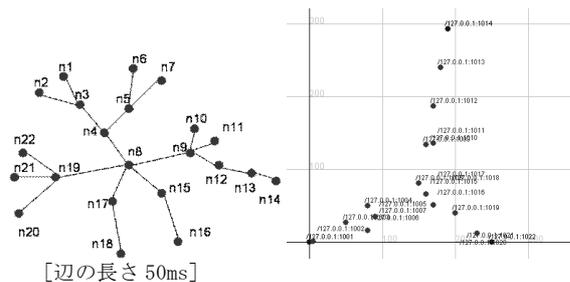


図3(左). 実験で使ったネットワークトポロジ  
(右). EDMAによって作成されたVE map

図2に PEP のアルゴリズムを示す. 自身の座標  $(x_i, y_i)$  の半径  $r_i$  内にノードがある場合, PEP の手続きを呼び出す. 参加するグループの候補  $G_{candidate}$  中のグループ内で一番遠いノードまでの距離  $l_{max}$  が, 一番近いグループの各ノードと遅延を測定し, 各ノードが定義した遅延 ( $D_i$ ) よりも遅延が小さければそのグループに参加すると共に座標  $(x_i, y_i)$  を正確に修正する. もし, 各ノードの遅延条件を満たさなかった場合, 他の候補のグループと遅延を測定する. もし, 参加可能なグループが見つからなかった場合, 自身の ( $D_i$ ) を大きくし VE map 上の半径  $r_i$  を大きくし, 参加可能なグループの探索を行う.

## 4 性能評価

本提案手法を J2SE v1.4.2 にて実装し, IBM X31 (CPU1.6GHz, Memory 512MB) 上で実行した. 遅延の最も少ないグループを探索する際に本アルゴリズムを利用した場合と, 総当りで探索した場合の時間差を測定した.

### 4.1 評価方法

評価実験を以下の手順で行った.

1. 図3(左)に示すトポロジをエミュレートし, ロビーサーバからノードのリスト, 基本測定ノードのリストを取得
2. 各ノードが RTT を測定し, 各ノード間の相対的な位置関係を VE map にプロットし VE map を作成
3. VE map 上の近くのノードと遅延を測定し, 各ノードとの相対的な位置関係を VE map に補正
4. 最も遅延の少ないグループを見つけ, そのグループに参加

### 4.2 測定結果

EDMA によって各ノード間の遅延時間の大きさを距離に置き換え, 作成した 2 次元のグラフ (VE map) を図3(右)に示す. 図4(左)に最も遅延の少ないノードを見つけ出すまでにかかった時間を示す. 図4(右)に本提案手法を使わずに総当りでグループを探した場合と本提案手法 EDMA を用いた場合のグループを探索した場合を比較したグラフを示す.

図4(左)ではグループ探索にかかる時間が 800ms から 3000ms までばらつきがある. これは VE map 上に座標を決める上で測定する 3 つのノードの遅延時間の差により発生したばらつきである.

図4(右)では総当りの方法と比較した場合, ノードの総数が 9 以下の場合, EDMA の方が座標を決定するための計算のオーバーヘッドがあるため近隣ノードを見つけ出すまで時間がかかっているが, それ以降はノードの遅延測定回数が少なく済み, 20 ノード以上の場合, 総当り手法より, 測定時間が 50%以下となる.

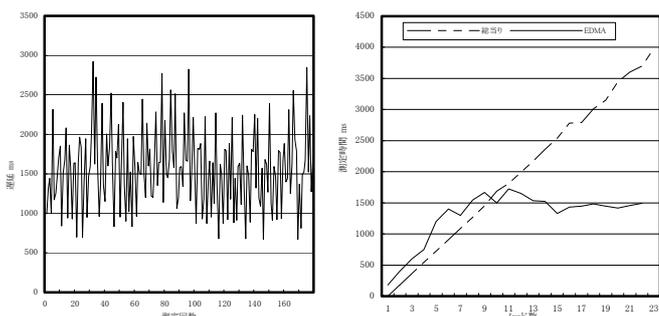


図6. (左)測定回数と最適グループ探索までの時間の関係(右)遅延が最も少ない近接ノードを見つけ出すまでの時間とノード数の関係

以上の結果より効率的に遅延の少ないノードを探索することができる.

## 5 関連研究

Pantel と Wolf<sup>3)</sup>は遅延がネットワークゲームに与える影響を示した. この論文によるとネットワークゲームを快適に行うには遅延は 50ms 以下, リアルタイムにゲームを行うには少なくとも 200ms 以下に遅延を抑える必要がある. Andersen ら<sup>1)</sup>はパスの断絶や状況の変化を検知し, 最適パスを素早く見つけ出す RON(Resilient overlay networks)を提案した. Hu と Liao<sup>2)</sup>はボロノイ図を用いて仮想空間上で隣接するノードとネットワークを組む手法を提案した. CAN<sup>4)</sup>や Chord<sup>5)</sup>は分散ハッシュを用いたシステムを提案した. CAN は N 次元トーラスでハッシュ空間を形成し, Chord では円状のハッシュ空間を用いたスキップリストという概念を使うことによって, 高速にオブジェクトの検索する. これらの方法はスケーラビリティに優れ, 負荷分散の効果がありインターネットのような大規模ネットワークに向いている.

## 6 まとめ

本研究では, P2P 環境でのネットワークゲームにおいて, 各ノード間の遅延を反映したネットワーク及びグループの作成の提案を行った. 提案方式では, 各ノード間の遅延をもとに作成した VE map を用いて各ノードの位置を決定し, グループを分けることにより, 効率良く遅延を決められた基準以内に制御できる. 今後はよりユーザ数の多い MMOG(Massively Multiplayer Online Game)へと研究の範囲を広げていく予定である.

## 参考文献

- 1) D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. In Proc. of SOSP, pp.131–145, Oct. 2001.
- 2) S.-Y. Hu and G.-M. Liao. Scalable peer-to-peer networked virtual environment. In Proc. of ACM SIGCOMM 2004 Workshops on NetGames '04, pp. 129–133, Aug. 2004.
- 3) L. Pantel and L. C. Wolf. On the impact of delay on realtime multiplayer games. In Proc. of NOSSDAV, pp. 23–29, May 2002.
- 4) S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In Proc. of ACM SIGCOMM, pp. 161–172, Aug. 2001.
- 5) I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In Proc. of ACM SIGCOMM, Aug. 2001.