

遅延測定に基づくネットワーク仮想環境グループ化の提案

十川 基[†] テープウィロー ジャナポン ニワット[§] 斉藤 裕樹[‡] 瀬崎 薫[§] 戸辺 義人[‡]

[†]東京電機大学 工学部 情報通信工学科

[§]東京大学 生産技術研究所

[‡]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は一般にサーバが必要で、電源、保管場所、メンテナンス等コストが発生する。また、プレーヤがサーバとして振舞い、他のユーザがそのプレーヤに参加するという方法ではスケラビリティに欠ける。これらの欠点を補うためにMOGをP2P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、どのようにトポロジを作成するか等の課題がある。そこで我々は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VE map (Virtual Environment map)を提案する。本稿では効率的なVE mapを構築するために粗く座標を推定した後に正確に位置を特定する2段階に分割する方法を提案する。

2 課題および前提条件

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し、ノードがネットワークの遅延を反映する地図としてVE mapを作成することとする。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こととする。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の存在を仮定し、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。

Measurement-based Peer-to-Peer Grouping for Networked Virtual Environment

[†]Hajime Sogawa

[§]Niwat Thepvilijanapong

[‡]Hiroki Saito

[§]Kaoru Sezaki

[‡]Yoshito Tobe

Dep. of Info. and Comm. Eng. Tokyo Denki University ([†])

Institute of Industrial Science, University of Tokyo ([§])

Dep. of Info. Systems and Multimedia Design Tokyo Denki Univ ([‡])

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pingコマンドに相当するコマンドでノード間の遅延を測定し、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:   gtest  $\leftarrow$  first group in Gcandidate
4:   repeat
5:     for all nodes in gtest 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:       $g_i \leftarrow$  gtest
11:      node  $i$  enters this group
12:     else
13:       gtest  $\leftarrow$  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

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 を大きくする．

3.4 具体例

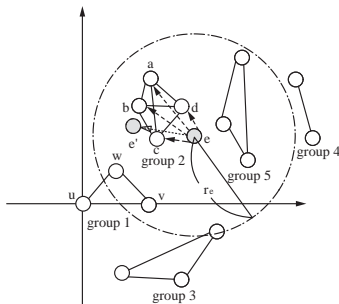


図3. VE map の例

図3に本提案の一例を示す．この例では5つのグループが存在し，同じグループに所属するノードは線で結ばれている．ノード u はこのゲームに最初に参加したノードである．次に参加したノード v の座標は $(0, r_{uv})$ である．ノード v は $(0,0)$ の円周上となるが，ここでは x 軸上の正の方向に置く．3番目にゲームに参加したノード w は既に参加している2つのノードと遅延を測定し r_{wu} ， r_{wv} より，座標を決める．ノード e がゲームに参加する例を説明する．ノード e がサーバから VE map と各ノード情報を取得，REP の手続きを行い，概略の位置を決める．次に PEP の手続きを行う．半径 r_e 内のグループ1と2が $G_{candidate}$ となる． $G_{candidate}$ のうち，ノード e は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2のノード a, b, c, d と遅延を測定する．グループ2の各ノードのノード間遅延の許容最大値を下回ればグループ2に参加し，座標を正確に修正し e' の座標となる．さらに新たにノードが追加された場合，上記手続きを繰り返す．

3.5 スーパーノード

拡張性を高めるためにグループごとの代表ノードとしてスーパーノードを提案する．これにより，今までの提案ではすべてのノードの中から無作為に3つのノードを選び出していたが，スーパーノードの中から無作為に3つのノードを選び出す．スーパーノードとなるノードは事前に定められた時間 (T_{join}) よりも多くゲームに参加してかつ各ノードは同じグループのすべてのノードと遅延を測定しその合計が最も小さいノード，若しくはグループに最初に参加したノードがスーパーノードとなる．スーパーノードは自身の所属するすべてのグループの座標などの情報をサーバに報告する．スーパーノードは，座標などの値に変化が無ければサーバに報告しない．スーパーノードがあることによりサーバへの座標のアップロードが少なくなり，拡張性が確保できる．

4 比較

スーパーノードを用いない手法とスーパーノードを用いた代表ノードを必要とする手法を比較する．データの転送回数 (T_x) とし， n ノードと m グループがあり $m \ll n$ と仮定する．スーパーノードがない場合の T_x は n ，スーパーノードがある場合の T_x は m となる．これよりスーパーノードの提案手法の方が制御パケットが少なく済み，より効率的な提案手法といえる．

5 関連研究

Pantel と Wolf³⁾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¹⁾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 Liao²⁾はポ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み合わせる手法を提案した．CAN⁴⁾や Chord⁵⁾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

本稿では，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 VE map をミドルウェアとして実装し，シミュレーションで評価を行っていく予定である．

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.