P2PMMOG ネットワークトポロジーによる遅延特性の評価

2007MI049 平山 佳典　　2007MI259 山原 雅也
指導教員　河野 浩之

1　はじめに

現在, MMOG(Massively Multiplayer Online Game) はクライアント・サーバモデルと Peer to Peer モデル (以下 P2P モデル) の 2 種類のネットワークモデルで運営されている． 多くの MMOG はコンテンツ管理やセキュリティの確保・データの一貫性の確保の容易さからクライアント・サーバモデルで運営が成されている．クライアント・サーバモデルはサーバへの負荷が大きく, イベント処理をするために遅延が発生する．遅延が発生することにより, キャラクターの位置座標やキャラクターの位置座標の一貫性の欠如の問題が発生し, ゲームが成立しなくなってしまう．

本研究ではソフトウェアシミュレータまたは, 実機を利用したテストベッド, 仮想ノードを利用したテストベッドを利用して, 遅延を減らすための負荷分散を行えるネットワークトポロジーのシミュレーションを行い, それらの遅延特性を観測し, ネットワークトポロジー考察をするものとする．

以下, 2 章でまず現状の MMOG ネットワークモデルと遅延による問題点を挙げる． 次に 3 章において,P2PMMOG のトポロジー提案を行い, 4 章でシミュレーションとその結果について考察し, 5 章でまとめとする．

2　MMOG のネットワークトポロジーと遅延問題

2.1　ネットワークトポロジー

既存の MMOG の P2P 通信形態として, ハイブリッド型 P2P を用いたものがある．その通信形態では, キャラクター情報やロビー (ルーム) 情報, 経験値やアイテムの獲得などの重要な部分を占める情報はサーバで通信を行い, それ以外の相手キャラクターへのダメージ判定などのゲームの対戦における部分は, プレイヤー同士で P2P 接続を行っている．

プレイヤーの接続間のネットワークトポロジーにはメッシュ型やリング型などが存在している．基本的なネットワークトポロジーにはどれも利点・欠点が存在し，大規模なネットワークや信頼性が求められるネットワークの場合, 複数のネットワークトポロジーを多層化やハイブリッドにすることがある．

2.2　遅延問題における影響

ゲームにおける遅延はパケットが送受信される間に生じる遅延時間以外にユーザの操作遅延なども含まれるが, 本研究では通信における遅延についてのみ考える．遅延発生により影響を受けるゲームと受けにくいゲームがある．FPS(First Person Shooter) や TPS(Third Person Shooter) のような即時性を要求されるジャンルのものは, 遅延で大きな影響を受ける．Pantel らの Real-Time Multiplayer Game[2] では遅延が及ぼす影響をプレイヤーの熟練度別に観測し, ユーザの体感に及ぼす効果を調査した結果, 500ms 以上の遅延はゲームを遊ぶ上で許容することができないという結果に至っている．逆に, ターン制のような即時性を要求しないボードゲームやシミュレーションは影響を受けにくいと言える．

実際に遅延が起こるとどのような現象が発生するのかということを FPS や TPS を用いて説明する．あるプレイヤーが他のプレイヤーに攻撃を当てた時，遅延が生じていない場合は見た目通りに当たり判定が生じる．しかし，遅延が発生している場合，相手に攻撃が当たった際に判定が生じずにしばらくした後に当たり判定が生じるなど，場合によってはその攻撃の当たり判定自体がなくなってしまう場合がある．このようなことが起こると本来なら倒せていたプレイヤーから攻撃を受けたり，当たったはずの攻撃が当たらなかったりなどゲームが成立しなくなってしまう．

既存の P2P を用いた MMOG と遅延によって出る影響を表 1 にまとめておく．

3　P2PMMOG に最適なトポロジーの提案

3.1　ネットワークシミュレータを用いたシミュレーションの提案

本研究では MMOG を P2P を用いて作る場合，どのようなネットワークトポロジーを用いるのが一番最適であるかということを遅延とノードの関係から考えていく．これを行っていく手法としてゲーム領域を小領域に分割し, その領域ごとにサーバとなるマスターノードを作成し, シミュレーションを行い, トポロジーの評価をしていく．領域を分割する方法として, ボロノイ分割を用いる．

表 1　P2P 型ゲームの一覧

タイトル	ジャンル	人数	遅延による影響
アラド戦記	アクション RPG	8 人	キャラクターへの攻撃判定の遅れ
BlackShot	FPS	16 人	キャラクターへの攻撃判定の遅れ
SD GUNDAM Capsule Fighter Online	TPS	8 人	キャラクターの位置ずれ

3.2　ボロノイ分割

ボロノイ分割とは, 施設などの最適配置を考える際に用いる分割方法である．ボロノイ分割の方法は, 複数個の母点との距離に基いて領域を分割した図で, 隣り合う母点間を結ぶ直線に垂直二等分線を引き, 各母点の最近隣領域を分割したものである．母点の配置の仕方により, さまざまな領域に分割することができる．

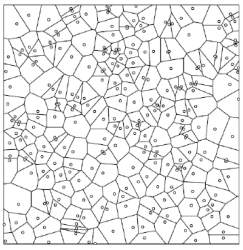

図 1　ボロノイ分割 (引用元:南山大学 情報システム数理学科 ボロノイ図 http://www.ms.nanzan-u.ac.jp/dep/voronoi.html)

3.3　Area Of Interest

Area Of Interest(以下 AOI) とは, あるノードが他のノードとの通信を行う範囲のことを指す．図 2 は★ノードの AOI の範囲を円として表している．★ノードの AOI 内の●ノード（AOIneighbor）は★ノードと通信を行いゲーム情報を交換し, そのゲーム情報をボロノイ分割された領域内にいるプレイヤーにゲーム情報を送信する．

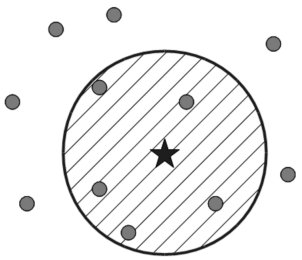

図 2　Area Of Interest

3.4　ネットワークシミュレーターの比較

本研究はゲーム領域をボロノイ分割により領域分割し，負荷分散を行った場合の遅延をネットワークシミュレーターを使用して計測を行う．今回の研究では OverlayWeaver, OMNeT++, ns-2 の 3 つのシミュレーターを候補とした．

OverlayWeaver は, アプリケーション開発やオーバレイのアルゴリズム設計などをサポートしているオーバレイ構築ツールキットである．このツールは, ルーティングアルゴリズムとして DHT(Chord, Kademlia, Koorde, Pastry, Tapestry) の実装を提供しており, 新規アルゴリズムの実装もすることができる．そして, 既存のアルゴリズムや新規に実装したアルゴリズムを試験, 評価, 比較するためのエミュレータも提供している．このエミュレータは数十万の仮想ノードを扱うことができ, 大規模エミュレーションによるアルゴリズム間の公正な比較を可能にする．

OMNeT++(Objective Modeler Network Testbed in C++) は，Andras Varga 等によって開発されたオブジェクト指向の離散イベント型のネットワークシミュレータであり, Unix と Windows が対応 OS となっている．C++ 言語のみで実装されているため, ユーザは提案手法を実装しやすいのが特徴である．さらに, GUI によるパラメータの設定, 確認などを行うことができるため, 評価環境を容易に構築することができる．また，リアルタイムでシミュレーション状況をアニメーション表示できることによって, 実装したプロトコルの動作把握やデバッグ処理が容易になるといったような特徴もある．

ns-2(Network Simulator version 2) は，カルフォルニア大学バークレイ校 (UCB/LBNL/VINT Project) によって開発された離散イベント型のネットワークシミュレータである．TCP, モバイルネットワークなどをサポートしており, インターネット上のプロトコル関連の研究に広く利用されている．

表 2 は以上のシミュレーターの実装環境や言語，特徴をまとめたものである．

本研究では, プレイヤーの Churn をパレートの法則を利用した ParetoChurn を利用するため, Churn の設定やゲーム領域・ユーザ数・グループメンバーなどのパ

表 2　シミュレータの比較

	OverlayWeaver	OMNeT++	ns-2
OS	J2SE を持つプラットフォーム	Windows,Unix	Linux
言語	Java	C++	C++, OTcl
特徴	・多くのアルゴリズムを実装 ・多くのルーティング アルゴリズムを実装 ・数十万のノードを扱える ・通信を実行時に可視化	・提案手法が実装しやすい ・パラメータ設定のみで シミュレーションが可能 ・GUI によるパラメータ設定, 確認 ・シミュレーション状況を アニメーション表示可能 ・実行中にパラメータ設定変更不可	・TCP, モバイルネットワーク等の シミュレーションが可能 ・イベントドリブン ・多くのネットワーク上の 機能が実装されている

ラメータを容易に行うことができるため OMNeT++ を用いることにする.

3.5　シミュレーションを行う P2P 環境

本システムの実行環境は以下のような環境で行う.

表 3　実行環境

CPU	Intel(R) Core(TM)2 Quad CPU
OS	Ubuntu 8.04
Memory	4GB
simulator	OMNeT++4.1
simulation model	OverSim

4　シミュレーション結果と考察

本研究では前章で延べた通り，OMNeT++ と OverSim を利用し，MMOG のゲーム領域をボロノイ分割して遅延を計測するシミュレーションを行っていく.

4.1　ネットワークシミュレーターのパラメーターの設定

本研究では OverSim のパッケージである Vast モデルを利用し，シミュレーションを行っていく．設定するパラメーターは以下のようにして行う.

area dimension はシミュレーションでのゲーム領域の大きさを表している．本研究では 1000m × 1000m, 2000m × 2000m, 5000m × 5000m, 10000m × 10000m の 4 種類の大きさのゲーム領域に設定し，シミュレーションを行う.

次に movement speed とは移動体速度で各ノードの移動速度を表している．本研究では世界中のプレイヤー数の 60 %のシェアを誇る MMOG である”World of Warcraft” で使用されている 5m/s を使用する.

Group size はゲーム領域内でのグループの数, つまりゲーム領域が分割されている数を表している．本研究ではこの Group size を 5，10，15，20，25，30，35，40，45，50 と変更してシミュレーションを行う.

AOI Width は各ノードの AOI の大きさを表している．ここでは 50 とした.

target overlay terminalNum はシミュレーションに参加しているノード数のことを言い，本研究では 50 と 100 の場合で計測を行った.

Churn はノードの参加, 離脱のことを表している．本研究では Churn のモデルとして ParetoChurn を使用する．Pareto Churn は MMOG やファイル共有に適しているモデルであり，本研究に適したモデルである.

simulated time は 10000s に設定する.

4.2　シミュレーション結果

4.2.1　node50 でのシミュレーション結果

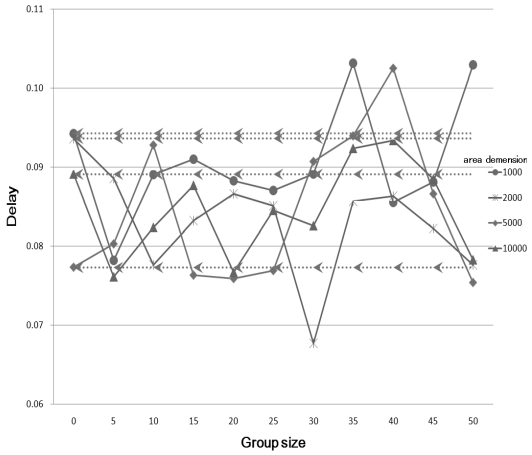

図 3　node50 でのシミュレーション結果

4.2.2 考察

前節のグラフを見ると node50 でのシミュレーション結果は Delay の数値が 0.06 から 0.11 の間の数値をとっており，2.2 で記述したように Pantel らの Real-Time Multiplayer Game[2] で述べられている 500ms 以上の遅延にはなっておらず，MMOG を行う際に問題は発生しないということがわかる．

さらには破線は area dimension が 0 の時を表しており，どのゲーム領域の大きさでも分割数が 15 から 25 の時に分割を行っていない場合よりも Delay が小さくなっている．

4.2.3 node100 でのシミュレーション結果

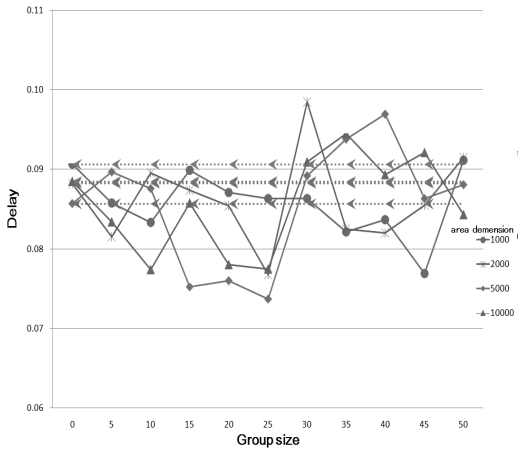

図 4　node100 でのシミュレーション結果

4.2.4 考察

前節のグラフを見ると node100 でのシミュレーション結果は Delay の数値が 0.07 から 0.11 の間の数値をとっており，node50 の時と同様に 2.2 で記述したように Pantel らの Real-Time Multiplayer Game[2] で述べられている 500ms 以上の遅延にはなっておらず，MMOG を行う際に問題は発生しないということがわかる．

さらには破線は area dimension が 0 の時を表しており、どのゲーム領域の大きさでも分割数が 15 から 25 の時に分割を行っていない場合よりも Delay が小さくなっている．

4.3 node50 と node100 での考察

図 3，図 4 を見ると，Delay の数値のバラつきは area dimension が大きくなるほど安定していく傾向を見ることができる．Delay の数値のバラつきは，node 数が 50 の場合より 100 の場合の方が Delay の数値が大きくなると考えていたが，node50 の時よりも node100 の時の方が Delay の数値が小さい傾向があるという結果を得た．

また，node の数値が大きいほど Delay の数値が大きくなると考えていたのだが，それぞれのグラフを見ると node50 の時よりも node100 の時の方が Delay の数値が小さい傾向があることがわかる．さらには node が 50，100 の場合はゲーム領域の分割数が 15 から 25 の時は分割をしていない状態よりも分割をしている状態の方が Delay の数値が小さくなるという結果を得ることができた．

5 まとめ

本研究では結果として，すべての数値がゲームを遊ぶ上で致命的な影響を及ぼすと言われている 500ms 以下の数値を示し，ボロノイ分割を行った場合，分割数が 15 から 25 の時に分割を行わない状態よりも Delay が小さくなるという結果を得ることができ, ボロノイ分割が有用であると言うことができた．しかし，今回のシミュレーションでは実世界でのプレイヤーの距離や各ノードのスペックの違いを考えず，ゲーム内のみでのシミュレーションを行ったものである．

そこで今後の課題としてゲーム内のみではなく実世界の環境も考慮し，シミュレーションを行い，計測をする必要があると考えられる．それにより実際の環境により近いシミュレーション結果が得ることができると考えられる．また，本研究ではノード数を 50 と 100 のみで行っているのでノード数を大きくしていった時に同様の結果を得ることができるかということやシミュレーション回数を重ねても同様の結果が得ることができるのかということも今後の課題である．

参考文献

[1] 宮地 利幸，三輪 信介，篠田 陽一, “ ネットワーク実験の支援技術, ” マルチメディア, 分散, 協調とモバイルシンポジウム, pp.797-807, 2009.

[2] Lothar Pantel, Lars Wolf,“ On the impact of delay on real-time multiplayer games. ”In Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video, pp.23-29. ACM Press, 2002.

[3] Michael Miller, “ P2P コンピューティング入門, ”株式会社 翔泳社，pp.229-309, 2002.

[4] 首藤 一幸, 田中 良夫, 関口 智嗣, “ オーバーレイ構築ツールキット Overlay Weaver, ” 情報処理学会論文誌：コンピューティングシステム，Vol.47, No.SIG12(ACS 15), pp.358-367, 2006.