

P2P ネットワークゲームのための 階層型遅延最適化ミドルウェアの提案と評価

金田 裕剛¹ 峰松 美佳² 齊藤 匡人² 間 博人² 徳田 英幸^{1,2}

¹ 慶應義塾大学 環境情報学部 ² 政策・メディア研究科

1 はじめに

ADSL や FTTH など大容量回線の普及により、ネットワーク越しにユーザ同士で遊ぶネットワークゲームが人気を集めている。ゲーム参加端末を直接接続し、ゲームを楽しむピア・ツー・ピア (P2P) 型のネットワークゲームは、システムの単一故障点となるゲームサーバを介さないことからその発展と普及が注目されている。

ネットワークゲームは各ユーザのゲーム状態情報を交換するための通信メッセージの交換方法 [2] として 2 つの手法が存在する。

1. タイムベース— 一定間隔でゲームの状態情報を受信することでゲーム状態の更新を行う。
2. イベントベース— 参加ユーザの操作毎にその操作の内容を送信することでゲーム状態の更新を行う。

また P2P 型においては中央で管理を行うノードが存在しないため、ユーザ同士を発見する手法が必要となり、その手法として主に 2 つ存在する。

1. ハイブリッド型— ユーザ同士の発見をある特定の認証サーバで行い、ゲーム開始時にユーザ同士が直接接続する形に切り替える。
2. ピア型— 探索的に周辺にパケットをブロードキャスト送信し、パケットに一番最初に応答したユーザに接続する。他のノードの情報はそのノードを経由して伝達される。

現在存在する P2P 型ネットワークゲームにおいては、状態情報の交換にイベントベースが用いられ、ユーザ同士の発見はハイブリッドで行われることが多い。また論理ネットワークトポロジには図 1 のようなメッシュ型モデルが利用される。

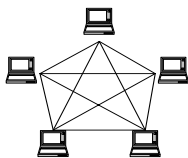


図 1: Mesh-P2P モデル

メッシュ型構造をとる P2P ネットワークゲーム (Mesh-P2P) ではノード数が増加するにつれて通信遅延の大きい経路を利用する可能性が大きくなる。イベントベースで同期を行うゲームの場合、操作内容が一部のゲーム参加者に到達するまでに時間がかかるためゲーム状態にずれが発生してしまう可能性がある。本研究ではゲームに参加するノード間で遅延を最適化するネットワークゲームのための階層型ミドルウェアを実装し、評価を行う。

2 ネットワークゲームにおける遅延の問題

2.1 遅延の影響

ネットワークゲームには従来の単一マシン用ゲームには存在しなかった問題が多々ある。大きな問題の 1 つにはネットワーク遅延のばらつき [1] が挙げられる。例えばある 3 人のユーザの通信経路のうち、1 つに致命的なネットワーク遅延があるとすると、イベントベースで実装されたゲームでは各ユーザの操作毎に操作内容が相手に送られるため、遅延の時間だけ遅れて参加者の操作を受け取ることになり、ゲームの状態にずれが発生し不平等な状態になる。これを回避するため、遅延が発生しうる十分な値までゲーム状態の更新を延期させることが考えられるが、これではユーザの操作は最大遅延間隔ごとにしか反映されない。このためネットワークゲームにおいてはより遅延の少ない通信経路を確保する必要がある。

2.2 Mesh-P2P における遅延の問題

Mesh-P2P においては、アプリケーションレベルマルチキャストによりユーザ同士が各々ユニキャストで接続する。この場合、遅延の大きい通信経路を利用する可能性がノード数が増えるにつれて高くなる。イベントベースで実装されているゲームでは、一部のノードでゲーム状態の整合性が取れない状況が起こりうる。

3 ANGEL の設計

Mesh-P2P における遅延の問題解決機構として ANGEL (Architecture for Network Games utilizing Eligible Leaders) を提案する。

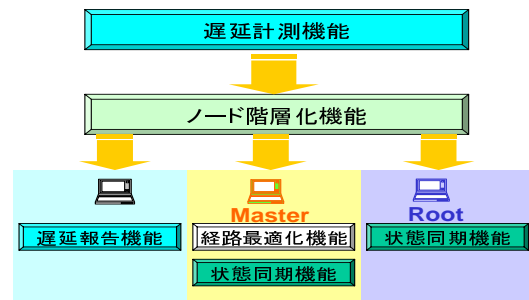


図 2: ANGEL 構成図

ANGEL の構成図を図 2 に示す。まずハイブリッド型によるノード同士の接続を行いメッシュを構築する。その後遅延計測機能により各ノードが接続しているノードに対して遅延計測を行い、そこからノード階層化機能により通常ノード (ON)・マスタノード (MN)・ルートノード (RN) に役割が分類される。分類されたノードのうち、ON は遅延報告機能により直接接続している通信経路の遅延を、遅延計測する間隔 (遅延計測間隔) に基づいて計測し MN に報告する。MN は ON からの遅延情報に基づいて経路ツリーを構築し、より低遅延な経路を発見した場合、経路切替メッセージを ON に送信する。これを持続的に行うことで経路の最適化を図る。

以下、各機能について詳細に述べていく。

- 遅延計測機能

Hierarchical Latency Optimization Middleware for Peer to Peer Network Games

Yugo Kaneda¹, Mika Minematsu², Masato Saito², Hiroto Aida², Hideyuki Tokuda^{1,2}

¹ Faculty of Environmental Information, Keio University

² Graduate School of Media and Governance, Keio University

5322, Endo, Fujisawa, Kanagawa 252-8520, Japan

E-Mail: {ichiriki,mine,masato,haru,hxt}@ht.sfc.keio.ac.jp

各ノードが直接接続しているノードとの遅延を計測し、そこから平均遅延と最大遅延を計算し、隣接するノードと交換を行う。

- ノード階層化機能

遅延計測機能によって計測された平均遅延を交換した結果、最も低い値を持つノードが自律的に MN として起動する。平均遅延が同じである場合は最大遅延から判断する。MN は起動した後、起動していることを知らせる間隔 (マスター確認応答間隔) に基づき確認応答を ON に送り、ON が MN が落ちたことを検知した場合、再度選出する。また MN が複数存在する場合、同様に MN 内から RN を選択する。

- 遅延報告機能

MN 起動後、それ以外のノードは ON として、直接接続している隣接ノードに対して遅延計測間隔で遅延の計測を行い、遅延の値を遅延テーブルに保持する。ON は遅延テーブルの情報を、遅延を報告する間隔 (遅延報告間隔) で MN に送信する。

- 経路最適化機能

MN が各ノードから集めた遅延情報から経路ツリーを作成する [7]。そしてより遅延の少ない経路を発見する間隔 (経路探索間隔) に基づいて経路ツリーを探索し、より遅延の少ない経路が発見された場合、経路情報から経路切替メッセージを送信する。各ノードはそれに基づいてアプリケーションレベルで通信経路を変更する。

- 状態同期機能

MN は各ノードのゲーム状態の一貫性を保つため、自身のゲーム状態を送信する間隔 (状態同期間隔) で ON に状態を送信する。MN が複数ネットワーク上に存在する場合は RN からのゲーム状態情報が MN にまず送信され、その後 MN から ON へ伝達される。MN は各ノードに対する遅延が平均的に少ないノードが選ばれているので、ゲーム状態の情報は全ノードへ最適化された遅延で配送される。

4 評価

4.1 評価環境

本システムのプロトタイプを RedHat Linux 9.0 において実装した。プロトタイプは遅延計測機能・遅延報告機能・経路最適化機能を有する。そして平面上でキャラクタが動き移動毎にその座標が各ユーザに送られるサンプルアプリケーションを GTK+ を用いて作成した。評価用にキャラクタの操作を自動化し、同一の移動パターンで行った。

評価環境は図 3 のように設定した。RedHat Linux 9.0 を載せた 3 台のノート PC を dummynet [5] に接続し、各経路で 200 ミリ秒、40 ミリ秒、40 ミリ秒前後の遅延を発生させた。遅延報告間隔を遅延計測間隔 5 回毎に固定し、遅延計測・経路探索間隔を共に 1~5 秒に変化させ 5 分間各 20 回ずつ計測し平均を取った。

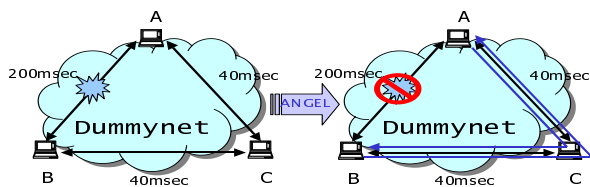


図 3: 評価環境と ANGEL 適応図

4.2 考察

経路収束時間と通常のシステムとの通信量の比較を図 4 に示す。経路収束時間は ANGEL 起動後、経路最適化が完了するまでの時間であり、通信量は実験に利用した 3 台の

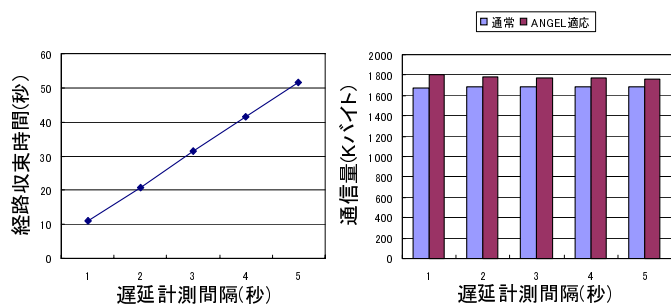


図 4: 経路収束時間と通信量

ノードにおける送信データ総量である。計測間隔が最長である 5 秒間隔において収束時間が平均 50 秒であり、平均ゲームプレイ時間が 15 分~30 分 [3] であることを考慮すると、接続時間の殆どの時間において遅延が最適化された状態で遊べるため実用的な範囲と言える。通信量は計測間隔が 1 秒の時に最大で 8% 増加する。しかし今回のサンプルアプリケーションではユーザ 1 人あたり毎秒約 1500 バイトのデータを送信しており、一般的なネットワークゲームに当てはめた場合、毎秒約 2800 バイト [4] 送信するため、通常のシステムに対する本システムの通信量の割合はさらに抑えられる。

5 関連研究

リアルタイムアプリケーションに関するアプリケーションレベルマルチキャストの最適化において以下の関連研究が挙げられる。

Narada [6, 7] は各ノードが接続しているノードへの遅延、帯域情報を計測し、それに基づいて自分を中心としたマルチキャストツリーを構築し、最適化を行う。Narada では各ノードがオーバーレイネットワークのトポロジによってマルチキャストツリー情報を別々に構築するのに対して、ANGEL では MN が一任して行う。従って ANGEL の方が参加しているノード数に対して経路算出の負荷がかかるノードが少ない。

6 まとめと今後の課題

本研究では、メッシュ型 P2P ネットワークゲームにおいて参加ノードを階層化し、役割を持たせることで遅延に基づく通信経路の最適化を行った。本システムによりネットワークゲームの公平性を維持するために重要である、メッセージ配送の遅延を低減することができた。今後の課題として最適な遅延計測・遅延報告・経路探索・状態同期間隔・マスター確認応答間隔・状態同期間隔の発見や、ネットワークにかかる負荷を減少させるために遅延報告の低減手法、また中継ノードの離脱から通信の回復が挙げられる。

参考文献

- [1] Jonathan Blow. A LOOK AT LATENCY IN NETWORKED GAMES. *Game Developer Magazine*, 1998.
- [2] Anthony R.Kurc Eric Cronin, Burton Filstrup and Sugih Jamin. An Efficient Synchronization Mechanism for Mirrored Game Architectures. In *Proceedings of ACM SIG NetGames*, 2002.
- [3] Wu-chang Feng Francis Chang. Modeling player session times of on-line games. In *Proceedings of ACM SIG NetGames*, 2003.
- [4] Josh Winslow Mark Claypool, David LaPoint. Network Analysis of Counter-strike and Starcraft. In *Proceedings of the 22nd IEEE International Performance, Computing and Communications Conference(IPCCC)*, 2003.
- [5] Luigi Rizzo. Dummynet: A simple approach to the evaluation of network protocols. In *Proceedings of ACM Computer Communication Review*, 1997.
- [6] Hui Zhang Yang-hua Chu, Sanjay G.Rao. A Case for End System Multicast. In *Proceedings of ACM SIGMETRICS*, 2000.
- [7] Srinivasan Seshan Yang-hua Chu, Sanjay G.Rao and Hui Zhang. Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture. In *Proceedings of ACM SIGCOMM*, 2001.