

SNEO : オーバレイネットワークを用いた ネットワークゲームのためのプラットフォームの提案

中村 陽一[†] 齊藤 裕樹[†] 戸辺 義人[†]

[†]東京電機大学 工学部 情報メディア学科

概要

近年、ネットワーク技術の発展により、コンシューマゲーム機や携帯ゲーム機などでも、さまざまなネットワークゲームが提供されている。現在、ネットワークゲームでは、主にクライアント・サーバ型と Hybrid-P2P 型の方式をとることが多い。しかし、これらの方式では、大規模なサーバの導入・運用など、多くの資源を必要とするため、ネットワークゲームを提供するのは容易ではない。そこで、我々は、P2P (Peer-to-Peer)通信を行うオーバレイネットワークを用いた、ネットワークゲームのためのプラットフォーム SNEO (non Server Network game Environment by using Overlay network)を提案する。我々のシステムは、従来、サーバで行っていたサービスをオーバレイネットワーク上の共通プラットフォームで実現させ、それを様々なゲームで共有して利用することができる。このシステムを使用することで、ゲーム制作者側は、ネットワークゲームの提供にかかるサーバやネットワーク資源を用意することなくゲームを提供することが可能となる。本論文では SNEO の概要と、ネットワークゲームにおいて特に重要となる対戦相手マッチング手法について提案する。

SNEO: Proposal of a Platform for Network Game Using Overlay Network

Yoichi Nakamura[†]

Hiroki Saito[†]

Yoshito Tobe[†]

[†]Department of Information Systems and Multimedia Design, Tokyo Denki University

Abstract

Recently, many network games have been developed by improvement of network technology in consumer game machines and portable game machines. Typical network game architecture is based on client-server model and Hybrid-P2P model. Because of heavy resource needs in servers, it is not easy to develop network games in game providers. We propose a platform based on overlay network which does not need the server for the network game, and develop a common platform in network game. Game provides can share the common platform in various games. Game providers become possible to produce games without considering about the load of networks and servers. In this paper, it proposes the outline of our platform, and the partner matching protocol which becomes essence in network games.

1. はじめに

近年、インターネットをはじめとするネットワーク技術や無線通信技術の発展に伴い、PC だけではなく、コンシューマゲーム機や携

帯ゲーム機などでもネットワークゲームが提供されている。ネットワーク上でのオンライン対戦ゲーム (TRPG, 麻雀, 将棋など) では、Hybrid-P2P 型や小規模なクライアント・サー

バ型の構成をとることが多い。Hybrid-P2P(Peer-to-Peer)型はゲーム自体の通信には P2P 通信を用いるが、Hybrid-P2P 型でも小規模なクライアント・サーバ型でも、対戦相手を探すためのマッチングにはロビーサーバを用いなければならない。したがってトラフィックや負荷の集中により、サーバの導入、運用に関してのコストの点で、個人だけでなく、企業であっても、ネットワークゲームを提供するのは容易ではない。また、応答性能が悪化したり、サーバに障害が起きた場合にゲームサービスが停止してしまうといった問題がある。

ネットワークゲームの一部には、提供側がサーバを用意する必要のないものもあるが、ゲームユーザ側が対戦する相手に IP アドレスを公開する、または、相手の IP アドレスを知らなければならないといった柔軟性に欠ける方式である。そのため相手を探すことは容易ではない。

そこで、我々はサーバを必要としないネットワークゲームのインフラストラクチャが必要であると考え、オーバーレイネットワークで構成されるプラットフォーム SNEO (non Server Network game Environment by using Overlay network) を提案する。このプラットフォームを用いることにより、ゲーム提供者側はサーバを導入、設置する必要がなくなりネットワークゲームの提供が容易になると考えられる。

2. オーバレイネットワークを用いたネットワークゲームのプラットフォーム

2.1 オーバレイネットワークの構造

オーバーレイネットワークを構築する方式には大きく、Unstructured オーバレイと Structured オーバレイの二つに分類される。Unstructured オーバレイは、ノード間のトポロジを規定していないオーバーレイネットワークである。目的とする資源を検索する際には、全ての隣接ノードに対してメッセージの転送を行う。そのため、遠くにある資源を見つけることが困難となるため、ノード数が増える

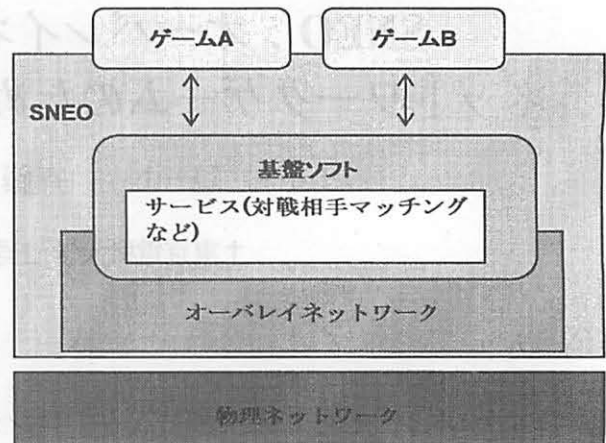


図1 SNEOの概要

に従い規模拡張性が問題となる。Unstructured オーバレイを用いた代表的なアプリケーションには、Gnutella[1]やWinny[2]などがある。一方、Structured オーバレイは、ノード間のトポロジとして、一定の構造を構成することで資源の検索や経路制御の効率を高めたオーバーレイネットワークである。目的とする資源を検索する際はトポロジにしたがって、隣接ノードにメッセージの転送をすればよい。そのためメッセージの転送が効率化され、規模拡張性が高い。このような方式には主に Chord[3], Tapestry[4], Pastry[5], CAN[6], Kademlia[7] などの分散ハッシュテーブル (DHT) が提案されている。DHT には、「高いスケーラビリティ」、「高速な検索、ルーティング可能」、「頻繁なノード参加・離脱に対応」、「すべてのノードと通信可能」という利点がある。そのため、従来はサーバで行っていたサービスを実現させるためには、Structured オーバレイが有用である。そこで、本システムでは Unstructured オーバレイではなく、DHT で構成される Structured オーバレイを用いる。

2.2 共通プラットフォームの設計

本研究では、図1に示すようなゲームアプリケーションから独立の基盤ソフトウェアにより、さまざまなゲーム共通のオーバーレイネットワークを構築するためのプラットフォームを開発する。

このオーバレイネットワークを複数のゲーム間で共有して利用することで、ゲーム制作者側はネットワークに関する部分を考えずに開発することが可能となる。これらの機能は基盤ソフトで提供する。そのためゲームユーザ側からは、基盤ソフトを用いれば、様々なネットワークゲームでも同様のサービスを受けることが可能となる。ゲームの通信方式は各アプリケーションの仕様により異なるが、オーバレイネットワークにおけるサービスとの通信は基盤ソフトを介して行う。

2.3 共有プラットフォームに求められる機能

共通プラットフォームに求められる機能は以下の通りである。

・対戦相手マッチング

ゲームユーザ側にとってネットワークゲームの重要な機能は、ネットワークを介して他のユーザと対戦することである。本研究では、サーバを用いることなく、マッチングを行う機能が必要である。マッチングのためには以下の3つの機能が必要である。

・固有の識別子で相手を探す機能

あらかじめ知っている人と対戦したい場合に利用する。

・無条件でランダムに相手を探す機能

無作為に選ばれた相手と対戦したい場合に利用する。

・レベルに応じた相手を探す機能

対戦成績により算出したレベル（ゲームの実力の数値化）を参考に「実力の近い相手」と対戦したい場合に利用する。

・共有オーバレイネットワークの構築

ゲーム毎に独立したネットワークを構築することは、コスト、運営、参加者数の確保など様々な問題が生じる。そこで、ゲーム間で共有して利用可能なネットワークが必要である。様々なゲームが同じネットワークを介することで、「どのようなゲームが存在するか」、「どのゲームが人気であ

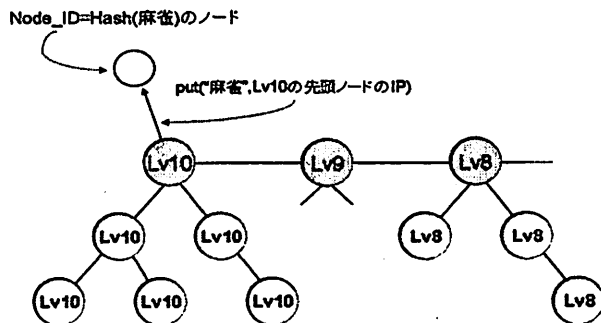


図2 レベルにより構造化されたツリー

るか」といった従来サーバで行っていた機能も提供することが可能である。

3. 対戦相手マッチング手法の設計

3.1 DHTを用いたALMツリーの設計

本章では対戦相手マッチング手法について述べる。本手法は既存のDHTを用いたStructuredオーバレイ上で行う。

本手法ではDHT上でアプリケーションレベルマルチキャスト(ALM)を応用する。DHT上でのALMはi3[8]やOverlay Weaver[9]などで実装されている。ALMではIPマルチキャストのように通信経路上のすべてのルータが対応している必要があるといった環境依存性がないため、制御や変更、サービスの提供が容易である。

3.2 実力を考慮したツリー構造

レベルに応じた相手をさがすことを可能とするためにオーバレイ上にゲーム対戦成績から算出したレベルを実力の指標とし、それにより構造化したツリーを設計した(図2)。そしてこのツリーを配送木としてALMを行う。

図2に示すツリー構造を説明する。まず同一レベルのノードのみでサブツリーを構築する。なお本稿ではサブツリーは2分木としている。ここでルートのノードのことを先頭ノードと呼ぶことにする。各レベルの先頭ノードは自分の±1のレベルの先頭ノードのアドレス情報を持つようにする。最高位のレベルの先頭ノードについて、NodeID=Hash(「ゲーム名」)のノードに先頭ノードのアドレスを格

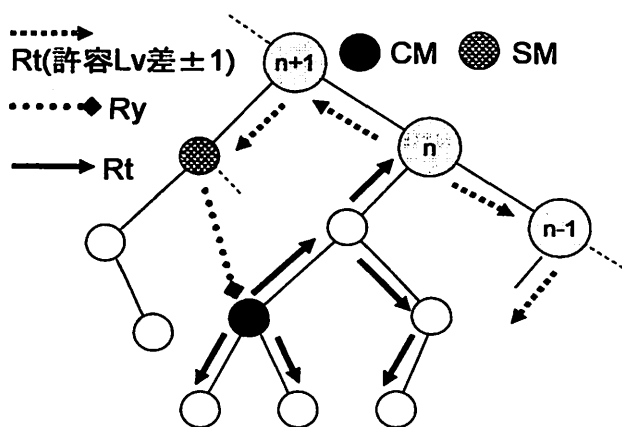


図3 メッセージの流れ

納する。図2ではLv10の先頭ノードのIPアドレスが NodeID=Hash(麻雀)に格納されている。

ゲームに新たに参加する際には、ハッシュ関数にゲーム名（ハッシュ値が重複しないようにゲーム名はユニークなものとする）を与え、その値を管理するノードに問い合わせ、Lv10の先頭ノードのアドレスを得る。Lv10の先頭ノードに接続し、目的のLvの先頭ノードまで辿る。次に目的のLvの先頭ノードにいき、子を進んでツリーに参加する。もし目的のLvの先頭ノードが存在しない場合、自らが先頭ノードとなり、上下のLvの先頭ノードに自分のアドレスを知らせる。

3.4 対戦相手マッチングプロトコル

マッチングの際のメッセージの流れを図3に示す。対戦相手を探そうとするノードが、自分の子、親ノードに対して対戦相手を探すためのメッセージを出す。それを受け取ったノードは子、親ノードに転送する。もしこのメッセージを受け取ったノードが対戦相手を待ち受けるノードであるなら、メッセージの送信者に対してゲームの対戦を承認する。

上記の対戦相手マッチング時において、ノード（ゲームユーザ）には3つの状態があり、状態は条件によって変化する。

• クライアントモード (CM)

ゲームに参加するために SM ノードを探し、接続するモード。

• サーバモード (SM)

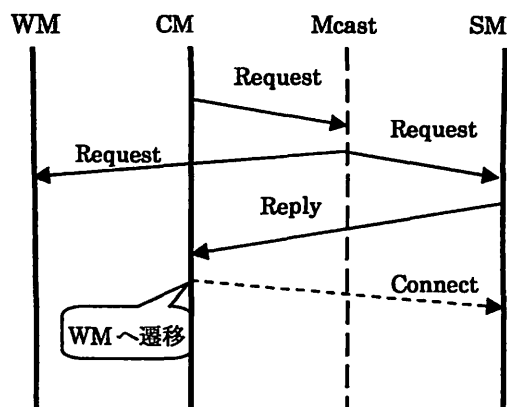


図4 基本的なメッセージフロー

ゲーム参加者からの参加要求を待ち、CM ノードから接続されるのを待つモード。

• ウェイトモード (WM)

CM ノードが SM ノードに接続した後と SM ノードがゲーム開始後になるモード。接続している CM, SM ノード以外からメッセージは転送のみする。

またノード間で通信されるメッセージは次の2つである。

• Request メッセージ (Rt)

CM ノードが自分の識別情報をマルチキャストし、SM ノードに対してゲームの参加を要求するメッセージ。また送る際、許容レベル差を含む。

• Reply メッセージ (Ry)

SM ノードが自分の識別情報を CM ノードにユニキャストし、ゲームの参加を承認するメッセージ。

各モードのノード間での基本的なメッセージフローを図4に示す。CM ノードが、参加したいゲームのツリーに参加し Request メッセージを送信する。ツリーに参加している SM ノードが Request メッセージを受信した後 Reply メッセージを送信する。WM ノードは応答しない。次に、Reply メッセージを受け取った CM ノードは SM ノードに接続し WM に遷移する。

さらに、以下に CM, SM, WM 各モードの

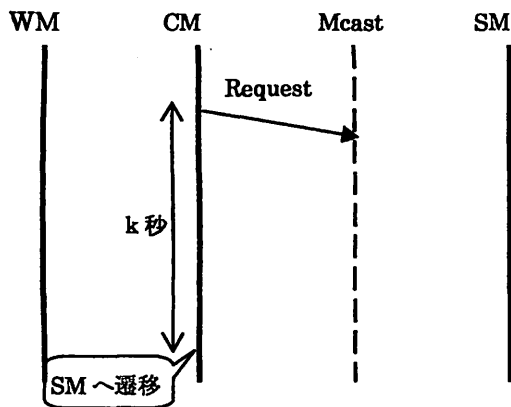


図5 SM ノードが存在しない場合における
メッセージフロー

動作手順を示す。

クライアントモード

図6にCMとWMアルゴリズムを示し、メッセージの流れについて以下に説明する。

1. 最初にノードは、自分の親、子ノードに Request メッセージ (Rt) を送る。
2. 受け取ったノード群は、自分が SM でない場合、その Rt を親、子ノードに転送する。SMであったなら Rt の送信元に Reply メッセージ (Ry) を送る。Reply を受け取った場合、WM に遷移する。
3. 先頭ノード(各レベルの一番上の節)は Rt に含まれる許容レベル差を参考に各レベルの先頭ノードに転送する(図7では許容レベル差 ± 1)。
4. 他の先頭ノードから Rt を受け取った先頭ノードは自分の子ノードに転送する。
5. 一定期間 Ry が送られてこなかったノードは SM に遷移する(図5)。

以上の手順により、許容レベル差を設定して Request を出すことで「自分とゲームの実力が近い相手と戦う」ことが可能となる。

サーバモード

SM のノードのアルゴリズムを図7に示し、以下に説明する。

1. CM モードからの Request メッセージを受け取り、対応する Reply メッセージを

```

--- CMmode -----
Mcast group = Game name
join (game group)
send _RequestMessage()
t = time which is CMmode
if(this.mode == CMmode & t < k){
    if(Reply message receive){
        connectd(SMnode)
        this.mode = WMmode
    }
}else{
    this.mode = SMmode
}
if(Request message receive){
    message is forwarded to (child & parent)-node
}

--- WMmode -----
if(Request message receive){
    message is forwarded to (child & parent)-node
}

```

図6 CM & WM アルゴリズム

```

while(this.mode == SMmode){
    if(Request message receive)
        send _ReplyMessage()

    member = Number of Game Players
    if(Number of CMnode connected == member){
        gameState = start
        this.mode = WMmode
        break;
    }

    t = time which is SMmode
    if(t >= Random time)
        if(Number of CMnode connected == 0)
            this.mode = CMmode
}

```

図7 SM アルゴリズム

送る。

2. CM ノードからの応答を受信する。
- 1,2 を、ゲーム側で設定された数のノードが揃うまで続ける。設定したノード数が揃い次第、ゲームを開始し、マッチングを終了する。以上が本手法の基本動作である。

もし、SM ノードが存在しない際に複数の CM ノードがほぼ同時の期間内に (図3中の k 秒間) Request メッセージを送信した場合、そのメッセージはどのノードでも受信されない。したがって、複数の CM ノードが SM に遷移し SM ノードが同時に複数存在することになる。これを回避するために、SM になってからランダムな期間、再び CM ノードからのメッセージが受信できなかった場合、さらに

CM に遷移し Request メッセージを送る。

4. オンライン対戦麻雀ゲームへの適用

本システムをオンライン対戦麻雀ゲームに適用した例を述べる。

麻雀ゲームの参加必要人数は 4 人である。そのため SM ノードは 3 つの CM ノードとの接続を行ったら、WM に遷移した元の CM ノードにマッチング終了を告げ、ゲームを開始する。その後ゲームが終わると、Request メッセージを出しマッチングを行う。その際、ほぼ同時期に元のゲームのメンバー達が Request メッセージを出すので再度同じメンバーが集まってしまうことも考えられる。これを避けるため、前ゲームでのメンバーからのメッセージは拒否できるようにする。友達と麻雀を楽しみたい場合には、あらかじめ、自分を特定できる識別子 (ユーザ ID など) を交換しておくか、仲間内だけのグループを作るという方法が考えられる。

5. 今後の課題

本論文ではオンライン対戦ゲームを対象とした共通プラットフォームを提案した。しかし、ネットワークゲームには、マルチユーザゲーム (MMOG (Massive Multiplayer Online Game) など) が存在する。現在、マルチユーザゲームは大規模なサーバでゲームの進行処理やリアルタイム処理を行っている。そのためサーバレス化が困難であるとされており、様々な研究が行われている ([10][11])。今後はマルチユーザゲームを考慮したネットワークゲームのためのプラットフォームを検討する。

6. おわりに

本論文では、オンライン対戦ゲームを対象とした、オーバーレイネットワークを用いたネットワークゲームのプラットフォームを提案した。DHT 上でのアプリケーションマルチキャストを応用し、従来サーバで行っていた対戦相手マッチングを、オーバーレイネットワーク上で行うための手法を述べた。この手法を

用いることで、従来ロビーサーバが必要であったゲームを P2P 通信のみで利用することができる。またゲームユーザを指定した対戦だけではなく、任意の同士や対戦相手を条件つきで探し出すことが可能である。

今後、プラットフォームの実装を行い、既存のゲームをプラットフォーム上に適用することによって有効性を検証する。

参考文献

- [1] J. Frankel, "The gnutella protocol specification v0.4 document revision 1.2," http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf, 2000.
- [2] 金子勇: Winny の技術. アスキー. 2005.
- [3] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F. and Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications, Proc. ACM SIGCOMM 2001, pp.149.160, 2001
- [4] Zhao, B. Y. et al.: Tapestry: A Resilient Global-Scale Overlay for Service Deployment, Journal on selected area in communications, Vol. 22, No. 1, pp. 41.53, 2004
- [5] Rowstron, A. and Druschel, P.: Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems, Proc. Middleware 2001, 2001
- [6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content Addressable Network. In ACM SIGCOMM 2001, pages 161.172, 2001.
- [7] Maymounkov, P. and Mazières, D.: Kademlia: A Peer-to-peer Information System Based on the XOR Metric, Proc. IPTPS'02, 2002
- [8] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, Internet Indirection Infrastructure, Proceedings of ACM SIGCOMM, August, 2002.
- [9] 首藤一幸, 田村良夫, 関口智嗣: オーバ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.ACS 15, 2006
- [10] Takuji Iimura, Hiroaki Hazeyama and Youki Kadobayashi, "Zoned Federation of Game Servers: a Peer-to-peer Approach to Scalable Multiplayer Online Games", In Proceedings of ACM NetGames 2004, August 2004.
- [11] B. Knutsson et. al. Peer-to-peer support for massively multiplayer games. In Proceedings of INFOCOM, 2004.