

遅延測定に基づくネットワークグループ化の検討

十川 基[†] 斉藤 裕樹[‡] 戸辺 義人[‡]

[†] 東京電機大学大学院工学研究科情報メディア学専攻〒101-8457 東京都千代田区神田錦町 2-2

[‡] 東京電機大学工学部情報メディア学科〒101-8457 東京都千代田区神田錦町 2-2

E-mail: [†] hajime@unl.im.dendai.ac.jp, [‡] {hsaito, yoshito}@unl.im.dendai.ac.jp

あらまし 近年、インターネットではネットワークゲーム、テレビ会議、VoIP などの遅延に敏感なアプリケーションが利用され、また従来のクライアントサーバ型から P2P ネットワークにて実現するための研究が盛んに行われている。しかし、P2P ネットワークにおいて遅延の影響を考慮したトポロジの形成についてはほとんど議論されていない。我々は P2P ネットワークを利用した遅延に敏感なアプリケーションの拡張性を向上させ、各ユーザが要求する遅延範囲内にグループ化を行うために、各ノードが遅延を測定し遅延を表現した VE map を作成する手法を提案する。本稿では VE map を元に P2P トポロジを形成するアルゴリズム EDMA(Efficient Delay Mapping Algorithm)とその実装、評価、結果を述べる。

キーワード P2P ネットワーク、遅延測定、オーバーレイネットワーク

P2P Grouping for the Delay Sensitive Applications

Hajime SOGAWA[†] Hiroki SAITO[‡] and Yoshito TOBE[‡]

[†] Information Systems and Multimedia Design, Tokyo Denki University 2-2 Kanda-Nishiki-cho, Chiyoda-ku, Tokyo, 101-8457 Japan

[‡] Department of Information Systems and Multimedia Design 2-2 Kanda-Nishiki-cho, Chiyoda-ku, Tokyo, 101-8457 Japan

E-mail: [†] hajime@unl.im.dendai.ac.jp, [‡] {hsaito, yoshito}@unl.im.dendai.ac.jp

Abstract Currently, the research for delay sensitive applications is to be used of the network game, video conference, and the distributed processing, etc. On the other hand, in the Internet to achieve it from a past client-server type on the P2P network is actively done. However, the formation of the topology that considers the impact of the delay on the P2P network is hardly discussed. Consequently, we consider that to make efficient network topology by measuring delay in P2P networks. In this paper, we propose a novel grouping algorithm that allows us each node measures the delay to group it within the range of the delay that the extendibility improvement of sensitive applications and each user demands from the delay using the P2P network, the delay map is designed to the algorithm that formed the P2P topology based on making and this map, simulated it, and evaluated it.

Keyword P2P Networks, Delay, Overlay

1. はじめに

近年、インターネットではネットワークゲーム、テレビ会議、VoIP などの遅延に敏感なアプリケーションが利用され、ネットワークゲームの分野では Star Craft[1], Ultima Online[2], War Craft[3] などの MOG (Multiplayer Online Game) がインターネットを通して行われている。これら MOG を実現させるためにサーバが必要となるが、インターネットを通じてゲームを行うためにコスト(電源、保管場所、メンテナンス等)が発生する。またプレイヤーの 1 台がサーバとして振舞い、他のユーザがそのプレイヤーに参加する方法を利用する MOG もある。この方法は低コストではあるが全てのデータを 1 台のプレイヤーが処理することとなり、

スケーラビリティに欠ける。またサーバとして振舞うノードとの遅延時間に応じて各プレイヤー間での不公平が生じる。こ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。しかし公平性やどのようにトポロジを作成するか、また P2P ネットワークに参加している異なる処理能力を持ったノードがどのようにデータを分担して処理するかなどの課題がある。また一貫性を維持するために受信したデータをすぐに反映せず、各ノードが一定時刻に送信したデータを各ノードが受信するタイミングで結果を反映し、描画する方法が考えられるがレスポンスが悪くなり遅延に敏感なゲームには向かない。プレイヤーによっては少ない遅延で対戦したいプレイヤーもいる一方で遅

延が大きいと、結果の反映と描画に時間がかかるプレイヤーもあり、プレイヤーの要求は各ネットワークの状態により異なる。しかし現在の P2P ネットワークは遅延を意識したトポロジコントロールを行わない。そこで我々は各ノード間の遅延を測定し、相対的な遅延時間を反映した VE map(Virtual Environment map)を用いてネットワークのコンディション、プレイヤーのポリシーに基づいて P2P のトポロジを形成する EDMA(Efficient Delay Mapping Algorithm)を設計し、実装、評価を行う。

2. P2P ネットワークのノードグループ化の課題

ネットワークゲームでは各ユーザの環境が異なるため、PC の性能(CPU,メモリ,グラフィックカードなど)ネットワークの帯域、遅延の影響などの各ユーザの環境により不公平が発生する。

L. Pantel と L. C.Wolf ら[4]はネットワークゲームの遅延の影響がユーザの操作性にどのような問題を与えるかを具体的に示した。遅延に敏感なリアルタイム性の強いゲームにおいてユーザは 50ms 程度の遅延はほとんどのユーザが気にならないが、100ms の遅延はユーザに不快感を与え、250ms では操作に著しく影響を与える。また、遅延時間はユーザによって許容できる範囲が異なることが述べられている。我々も図 1(a)に示す TANKZERO[5]というリアルタイム性の強いネットワークゲームを作成し図 1(b)に示すようにプレイヤー間に遅延発生器を用いて擬似的にインターネットを模擬し高校生及び大学生 20 名で同様の実験した結果を表 1に示す。

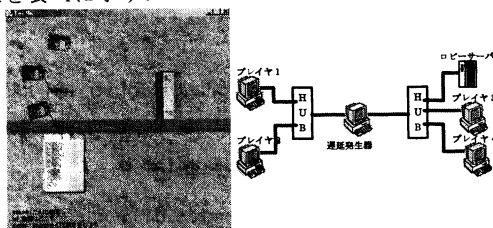


図 1 (a)TANKZERO の画面例 (b)ネットワーク構成図

表 1 遅延の影響

遅延(ms)	考察
50	遅延の影響は気にならない
100	ほとんどの人が遅延により操作に影響があると感じる 通常の動作では遅延はあまり気にならない 接近戦でインタラクションが多い場合不利になる場合がある
150	遅延の影響で操作に影響を及ぼす 正確にコントロールするにはこの遅延環境での慣れが必要
200	遅延の影響が操作に著しく影響を及ぼす
250	遅延の無いプレイヤーと操作性に圧倒的な差があり、対戦にならない

表 2 遅延の比較

(東京電機大学から各大学への ping 応答結果)

宛先	RTT(ms)
www.dendai.ac.jp	6
www.u-tokyo.ac.jp	10
www.hokudai.ac.jp	39
www.stanford.edu	140
www.berkeley.edu	144
www.fudan.edu.cn	212

このように、遅延の影響は操作性、描画に影響を与え、公平に対戦することができないことがわかる。

インターネットにおける遅延は以下の原因の二つが考えられる。

・ NIC の影響

使用する NIC により帯域、応答速度などに差があるため同じネットワークセグメントを利用したとしても RTT が異なる。

・ インターネットのトポロジによる遅延

インターネットでは物理的な距離と接続経路によって各ノードとの遅延が発生する。表 2 に示すように地理的な距離のみならず、ネットワークの物理的な接続経路により、アメリカよりも中国のほうが遅延時間は大きい場合がある。またネットワーク機器や通信路の状態により遅延にゆらぎが発生する。

以上の様に各ノードの遅延はユーザに悪影響を与え、その遅延は使用するデバイスや端末間の距離や使用する NIC により異なる。しかし、遅延の影響をネットワークトポロジによって改善する議論はほとんどされていない。そこで我々は P2P ネットワークを用いた場合の遅延の影響を少なくするアルゴリズムを検討する。

P2P ネットワークでは従来のクライアント/サーバモデルと同様にデータを管理するピアが複数存在し、選択可能なグループ数が増加する。

これにより遅延の大きい、遠くのサーバにアクセスせざるを得なかったノードは近くに存在するノードにアクセスすることにより、快適に操作することができる。また使用しているインタフェースが遅い場合はその環境と似たノードとグループ化を行うことにより、遅延の影響による公平性を維持することができる。

そこで本研究では P2P 型のネットワークで、各ユーザが要求する遅延時間の範囲、参加人数などのパラメータをもとに最適なグループを探索するアルゴリズムを考案する。

上記目的を実現するために各ノードとの遅延時間を全て測定すると、

測定時間 $T = \text{ノードとの遅延測定時間 } t \times \text{ノード数 } n-1$ となり、スケーラビリティに欠ける。そこで少ない測定回数で各ノードとの相対的な位置関係を把握し、各ピアの相対的な位置関係を予測する手法が必要である。

設計指針

本提案手法では、以下の 3 つを設計指針とする。

- ・ 各ノードの要求するグループの探索
- ・ 各ノードの要求する遅延範囲が異なることを前

提とし、ノードごとに最適なグループを探索する

- ・遅延が最も少ないグループの探索
- ・ユーザが許容する遅延範囲内に収まるグループの探索
- ・参加人数を重視したグループの探索
- ・効率的なグループ探索
グループの探索する最適なアルゴリズムを考案
計算コストが少ない方法を提案する
- ・探索グループの最適化
インターネットの遅延時間は時間とともに刻一刻と変わるため、以前測定した遅延時間が時間と共に異なる場合がある。そのため、遅延を定期的に再測定し、最適グループを常に探索する。

3. 遅延地図を用いたグループ化の提案

本研究ではユーザが指定した遅延時間内に収まるグループを素早く見つけ、グループ化する方法を提案する。グループ化するにあたり各ノード間の遅延時間を元に作成した地図を用いて、各ノード間の相対的な位置を把握し、各ノード間の相対的な遅延時間をユークリッド距離に対応づけ、各ノード間の相対的な位置関係を把握する。まず図 2(a)のネットワークの場合を考える。縦軸、横軸に遅延時間を取り、ノード間の相対的な遅延時間を表している。現在ノード 4 が各ノードと遅延を測定し、遅延時間を示した図を図 2(b)に示す。ノード 4 から他のユーザまでの遅延時間は 60ms で、その交点に遅延の地図にプロットすることとなるが、2 次元平面では交点は存在しない。そこで図 3 のように 3 次元空間上にプロットすることで正確に各ノード間の遅延時間を表すことが可能となるが、ノード数がさらに増えるにしたがって 3 次元空間上でも表現することが困難になり、結果的に各ノード間の遅延を正確に示すには n 次元で表現することとなる。しかし n 次元空間で各ノードの位置関係を正確に計算する場合、計算コストが膨大になってしまう。また、traceroute などを用いて途中の経路を測定し、経路上のネットワークの地図を形成する方法も考えられるが、現在のネットワークはセキュリティの観点から ICMP Echo Reply を行わないネットワーク機器が多く、また各ネットワーク機器を元に経路図を作成する場合はスケラビリティにかけ、有効な手段とは考えられない。そこで、我々は近似的に各ノード間の相対的な遅延時間を 2 次元平面状にプロットし、その後ノードごとに位置を再測定し、ノードごとの遅延地図を修正することにより座標の精度を上げる方法を提案する。

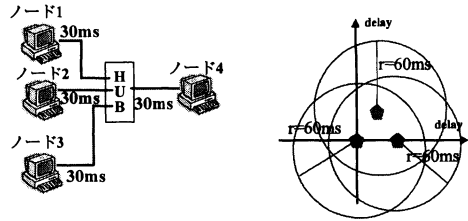


図 2 (a)ネットワーク構成図(b)2次元での遅延地図

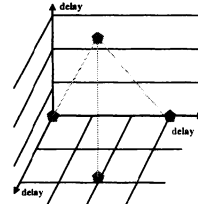


図 3 3次元で表した遅延地図

3.1. 前提条件

本提案を行うにあたり、前提条件を下記に記す。

- ・基本測定ノード
グループに参加しようとしているノードが各ノード間の相対的な位置関係を決定するために、遅延を測定する基準のノード
- ・ロビーサーバ
基本測定ノード、グループに所属している全ノードの IP アドレス、座標などを管理する
- ・参加ノード
グループに参加しようとしているノード。ロビーサーバの IP アドレスを知っており、まずロビーサーバにアクセスし、基本測定ポイント、各ノードの IP アドレスなどを取得する。

3.2. 遅延地図の作成アルゴリズム

効率的なグループ化を行うために VE map (Virtual Environment map)とグループ化アルゴリズム EDMA (Efficient Delay Mapping Algorithm)を提案する。

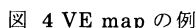
- ・VE map
横軸、縦軸共に遅延時間を表し、各ノードとの相対的な遅延時間を 2 次元平面上に表したグラフを提案する。図 4に VE map の例を示す。ユークリッド空間の距離が遅延に相当するため、ノード間の距離が遅延に相当する。
ノード n_1 の座標が $n_1(x_1, y_1)$ 、ノード n_2 の座標が $n_2(x_2, y_2)$ の時

$$\text{ノード } n_1 \text{ と } n_2 \text{ の遅延時間 } P_{n_1, n_2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

となる。

以下に 2 種類の VE map を提案する。

- ・Master VE map
ロビーサーバが管理する VE map。参加ノードが最初に取得する各ノードとの相対的な遅延時間を表す。
- ・Refined VE map
各ノードが管理する VE map。ロビーサーバから取得した Master VE map を元に参加ノード

• **EDMA**

- ・より精度の高い VE map の生成、修正

1. 参加するノードはロビーサーバから基本測定ノードの情報と参加ノード情報(座標, グループ, IP アドレス)を取得する.
2. 参加ノードが 3 つ以下の場合, 基本測定ノードとなりその他のノードからの測定に応答するノードとなる.
3. 基本測定ノードと遅延を測定し, Master VE map 上の位置を決める.
4. VE map 上の座標の位置から他のノードと遅延を測定し, 他のノードとの位置関係を Refined VE map 上に修正する.
5. いずれのグループとも条件が合わなかった場合はユーザに許容遅延範囲を大きくする旨通知するか新しいグループを作成するかをユーザに確認し終了する.

```

1: Procedure EDMA (delay_capacity; number_of_member);
2: begin
3:   Get Master VE map from Lobby server;
4:   if n = 0 then//n is number of node
5:     plot (0,0) on VE map;
6:   else if n = 1 then
7:     measure delay to the another node;
8:     plot (n1, 0) on VE map;
9:   else if n = 3 then
10:    measure delay to n0 and n1; plot on VE map;
11:   else //n>3
12:    measure delay to basic measurement nodes;
13:    plot on VE map;
14:   while 1 do
15:     begin
16:       measure delay to basic measurement nodes;
17:       refine coordinate on Master VE map;
18:       call refine phase for Refined VE map ;
19:       join with delay_capacity && number_of_member
20:     group
21:   end;
22: end;

```

・ 基本測定ノードの要件

- ・ 有線接続/遅延が少ない

遅延はネットワークグループ化の課題で示した様に使用する NIC によって遅延が異なる。特に無線デバイスは電波の受信状況により遅延や帯域が頻繁に変わるため、有線接続のノードを優先する。

- ・その他の基本測定ノードと地理的に離れているより正確に三点測量するには他の基本測定ノードと離れていた方がより高い精度で VE map にマップすることができる。

- 一定時間以上ネットワークに接続している基本測定ノードはその他のノードが遅延を測定するために必要なため、ネットワークにできるだけ長く接続しているノードが望ましい。

・ アルゴリズムの動作例

EDMA の動作例を図 4 を用いて説明する。ノード 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 3 つの基本測定ノードと遅延を測定し、概略の位置 (x_i, y_i) を決める。最初のノードが参加する場合、そのノードの座標はノード u のように $(0,0)$ となる。2 番目に参加するノードは 1 番目に参加しているノードとの遅延時間の円周上の座標となり x 軸上にプロットする。ノード v がその場合である。

ノード 3 つ目以降から各ノードとの相対的な距離をもとに三点測量により座標を決定する。ノード e は基本測定ノード u, v, w と遅延を測定し、そしてノード e から一番近いノードはノード d と遅延を測定し、座標を修正する。

4. 実験

本章では本提案手法の有効性と実用性を確認するために、Linux、Windows マシンを 8 台用いてネットワークを組み、EDMA を用いた場合と総当たりによりグループを決める場合を比較する。

4.1. 実験環境

遅延発生器の OS は Red hat Linux 9.0, 遅延をエミュレートするソフトウェアとして NIST NET を利用した。各ノードは Red hat Linux 9.0 及び Windows 2000 上に本提案手法の VE map, EDMA を J2SE v1.4.2 で実装した。図 6 にネットワーク構成図を示す。

遅延発生器は各ネットワークセグメントの間に設置し、セグメントを跨いだ通信を行う際には100msの遅延を発生させ、同一セグメント内は30msの遅延を発生させた。またノード4,5,6はポート番号を変更することにより数10台の参加ノードをエミュレートする。この構成により基本測定ノード3台、参加ノード3台から

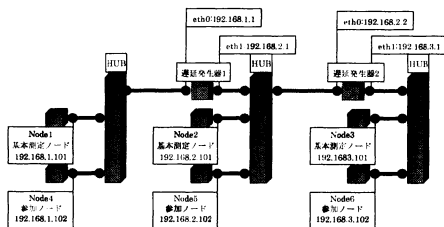


図 6 実験ネットワーク構成図

60 台までエミュレートし実験を行った。

4.2. 実験内容

評価実験を以下の手順で行った。

- (1) 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する
- (2) ノード 1, 2, 3 が基本測定ノードとなり各ノードとの相対的な位置関係を示すための被測定ノードとなる。
- (3) ノード 4, 5, 6 はグループを組むための参加ノードで遅延を測定し，そしてグループ化を行う。
- (4) 各ノードが基本測定ノードのノード 1, 2, 3 と遅延を測定し，各ノード間の相対的な位置関係を VE map に反映する。
- (5) 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 に補正する。
- (6) ノードの要求する遅延範囲内にあるグループを見つけそのグループに参加する。今回行う実験では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する。

4.3. 測定結果

30 の参加ノードとのノード間の遅延を測定し，ノード間の相対的な遅延時間を示した VE map を図 7 に示す

・ 効率性の評価

本提案手法である測定ポイントと遅延を測定し，相対的な位置関係を元にグループを探索した場合と総当りでグループを探索した比較を表 3 に示す。

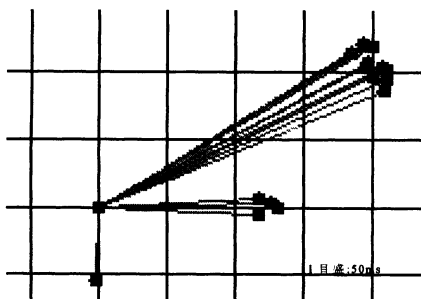


図 7 実験で取得した VE map

表 3 最適グループの探索に関する比較

	本提案手法	総当り
探索時間 ms (最小/平均/最大)	1500/1680/1954	100/2210/4641
探索回数 (最小/平均/最大)	4/4/4	1/10/30

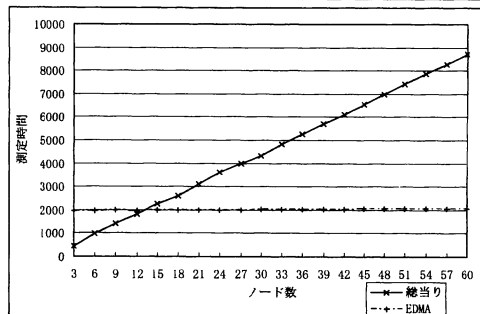


図 8 ノード数と測定時間の関係

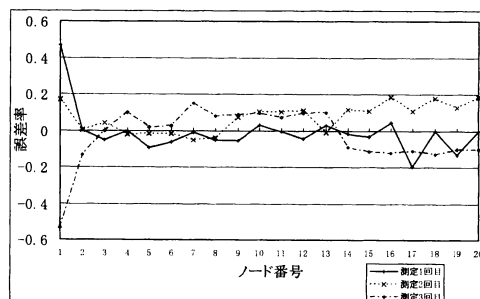


図 9 VE map での誤差率

総当り手法では遅延時間が 50ms 以下のノードを見つけるまでの時間を測定した。ランダムに探索するため，1 回で見つかった場合は，素早く終了するが，見つからない場合は全てのノードと遅延を測定するため，時間がかかる。一方，本提案手法の EDMA では 3 つの基本測定ノードと遅延を測定し，一番近くに存在するノードと遅延測定するため合計 4 回で探索することができ，効率的かつ安定してグループが見つけれることが分かる。また，参加ノード数を 60 まで増やした時の探索にかかる時間を比較したグラフ図 8 に示す。総当り手法では遅延測定回数がノード数に比例するが本提案手法は 2000ms 付近で安定していることがわかる。

・ 精度の評価

本提案手法では基本測定ノードと遅延を測定した後，座標の精度を上げるために，近いノードから順に再度遅延を測定し，VE map の精度を上げる。この手法が有効なのかどうかを検証するために以下の方法で座標の精度の検証を行う。

VE map 上で予想されたノード n との遅延時間 P_{delay_n}

$$P_{delay_n} = \sqrt{(x_{own} - x_n)^2 + (y_{own} - y_n)^2}$$

x_{own} : 自ノードの x 座標

x_n : ノード n の x 座標

y_{own} : 自ノードの y 座標

y_n : ノード n の y 座標

実際に測定したノード n との遅延時間 T_{delay_n}

$$\text{誤差率 } E_n = \frac{T_{delay_n} - P_{delay_n}}{T_{delay_n}}$$

1 度目の測定により生成された VE map をもとに予想された遅延時間と実際に測定した遅延時間との誤差率のグラフを図 9 に示す。

測定 1 回目、ノード 1 の誤差率が 0.47 となっているが、これはノード 1 との予想された遅延時間は 54ms、測定した遅延時間は 34ms と遅延時間そのものが小さいため、誤差率に大きく影響した。その他のノードとの誤差率は 0.2 以内に収まっており、各ノード間の相対的な遅延時間をおおよそ表現できていることが分かる。今回の実験では誤差率が最大 0.5、平均 0.1 発生し、2 度測定し VE map の座標の精度を上げる必要性が見受けられる。誤差が発生する原因として JAVA のタイマは JVM の仕様上、10ms の誤差を含むことが起因と考えられる。また、3 点測量はガウス・ジョルダン法を用いて行っているが、各ノードの遅延測定値に誤差が含まれるため、座標に誤差が含まれ、VE map 上で想定される遅延と実際の測定値が異なってしまうことが原因としてあげられる。また基本測定ノード間が十分に離れていない場合、三点測量がうまく機能せず、正しくプロットできないことも本実験を通して確認することができた。その一方で、粗い地図ではあるものの総当りに比べると効率的に機能していることが確認することができた。

5. 関連研究

Andersen ら[6]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。パスの断絶を素早く発見することはできるが P2P のトポロジをグループ化することについては述べられていない。Hu と Liao[7]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。この手法は仮想空間の状態をネットワークトポロジに反映させるため、ネットワークの状態は特に考慮されていない。CAN[8]や Chord[9]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が、P2P ネットワークの遅延に関しては述べられていない。

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効率的なグループ化の提案を行った。本提案方式では、各ノード間の遅延をもとに VE map を作成し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する手法を提案した。また効率的にユーザが要求するグループを見つけ出すために基準ノードと遅延を測定し、各ノード間の相対的な遅延時間を VE map にプロットし、遅延時間が最も短いグループを見つけ出すまでにかかる時間を測定し、本提案が効率的な手法であることが確認した。今後は参加ノードの数を増やし、ネットワークをより複雑にした上でのスケーラビリティの確認及び、より効率的なアルゴリズムの考案を行っていく予定である。

文 献

- [1] StarCraft. <http://www.blizzard.com/starcraft/>.
- [2] Ultima online. <http://www.uo.com/>.
- [3] World of Warcraft. <http://www.worldofwarcraft.com/>.
- [4] L. Pantel and L. C. Wolf. "On the impact of delay on realtime multiplayer games," In Proc. of NOSSDAV, pp. 23–29, May 2002.
- [5] TANKZERO. <http://www.unl.im.dendai.ac.jp/ageha/tankzero/>.
- [6] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. "Resilient overlay networks," In Proc. of SOSP, pp. 131–145, Oct. 2001.
- [7] S.-Y. Hu and G.-M. Liao. "Scalable peer-to-peer networked virtual environment," In Proc. of ACM SIGCOMM 2004 Workshops on NetGames · 4, pp. 129 · 33, Aug. 2004.
- [8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. "A scalable content-addressable network," In Proc. of ACM SIGCOMM, pp. 161 · 72, Aug. 2001.
- [9] 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.