

# ネットワークゲーム研究の理論 と実際

東京大学 大学院情報理工学系研究科

川原 圭博

# 本発表の内容

- 目的
  - ゲーム開発に用いられるネットワーク技術について、研究者コミュニティで共有されている諸問題について紹介する
- カバー内容
  - ゲーム開発に使われるネットワーク技術の分類
  - MMORPGに見るスケーラビリティ向上技術
  - ネットワークのことを知る技術
  - ユーザのことを知る技術
  - 実空間情報を取り入れたゲーム

# ゲームにおけるネットワークの使われ方

## ゲーム自体の配布

- ソフトウェア販売
- 本体またはアプリケーションのアップデート

## 課金管理

- プレイ時間による従量制課金
- アバターやアイテムの課金

## 複数ユーザーとのプレイ

- オンラインで得点の競争
- インターネットで見ず知らずの人プレイ
- アドホックで近くに居る人とプレイ

# ネットワークは土管か？

-プログラマ視点-

```
wclient(String host, String file){
    try {
        s = new Socket( host , 80 );
        in = new DataInputStream(s.getInputStream());
        out = new PrintStream(s.getOutputStream());

        String str= null ;
        out.println("GET " + file + " HTTP/1.0");
        out.println();

        while((str = in.readLine()) != null){
            System.out.println(str);
        }

    } catch (Exception e){
        System.err.println("Exception(new 3) : "+e);
    }
}
```

# ネットワーク技術者視点

-あっちこっちに課題がいっぱい!!-

アクセス網:  
エラー多し



(携帯)端末

携帯基地局



負荷の集中



サーバ

キャリア  
ネットワーク

ISP

Internet

Best  
Effort

読めないユー  
ザの挙動



Wifi

複数の管理ドメインにまたがるためトラブルへの対応が難しい

# オンライン対戦 ゲームの構成技術

# オンライン対戦ゲーム研究の興り

## Networked Virtual Environments

### ■ Networked virtual environments (Net-VE)

- 複数ユーザがリアルタイムに協調する
- リアルタイム性
- 没入感

*(Sandeep Singhal, Michael Zyda, "Networked Virtual Environments: Design and Implementation", Addison-Wesley, Sep. 1999)*

### ■ 応用

- 軍事訓練
- 作業の練習
- 教育
- オンラインゲーム

NPSNET



STOW



DIVE

Age of Empires  
(c) MicrosoftPAW2  
(c) SONYFinal Fantasy XI  
(c) SQUARE

# ゲーム構成データと特徴

	帯域(bits/s)	性質	バースト性	遅延への 敏感さ	ジッタへの 敏感さ	エラーへの 敏感さ
UDP avatar	12K x n (30fps)	Interactive Real-time	Constant	Y	Y	N
UDP audio stream	64K x n		Brief	Y	Y	N
UDP video stream	10M (2-way only)		Constant	Y	Y	YN
UDP stream With Playback	depends	Non- interactive Real-time	Constant	Y	N	YN
TCP control data	7K x n	Reliable	Brief	YN	YN	Y
TCP bulk data	depends	Best Effort or Deadline Delivery	Sustained burst	N	N	Y



# 最近のゲームの例

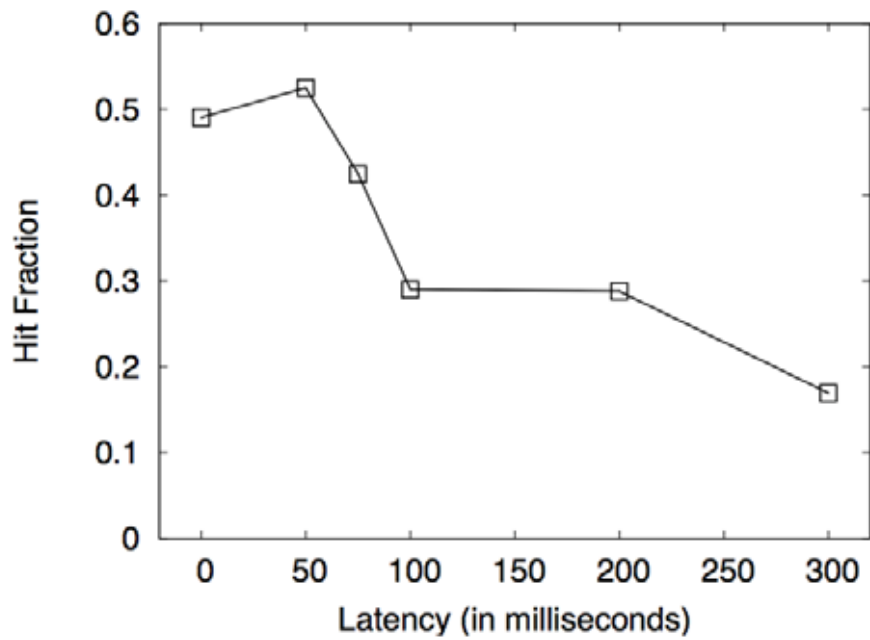
Game	Bandwidth (kbits/s)	Pkt Size (bytes)	Inter-Pkt (msec)
Warcraft	5	49	200
Madden NFL	14	77	75
Unreal Tournament	67	75	45
Second Life	775	1027	9

- Second Lifeはほかのゲームに比べてネットワークの要求が高い
  - Bandwidth use      10x-100x
  - Packet sizes        15x-20x
  - Packets sent        3x-20x
- Traffic Analysis of Avatars in Second Life by James Kinicki and Mark Claypool
  - NOSSDAV 2008

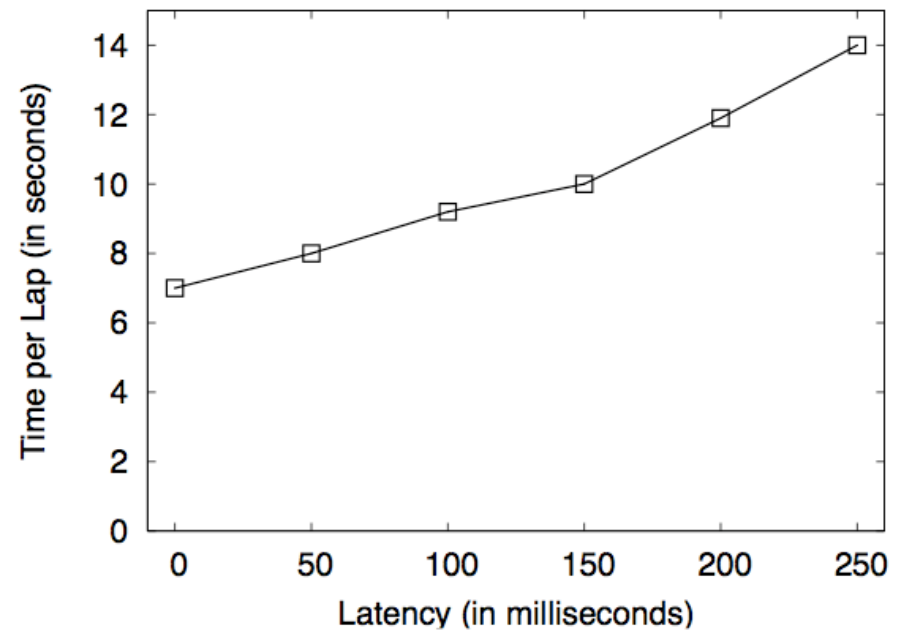
# ネットワークのパラメータ

- ゲームの善し悪しに最も影響するパラメータは遅延
- On Latency and Player Actions in Online Games
  - By Mark Claypool
- ネットワークゲームの各種遅延を精査
  - 2003-2006

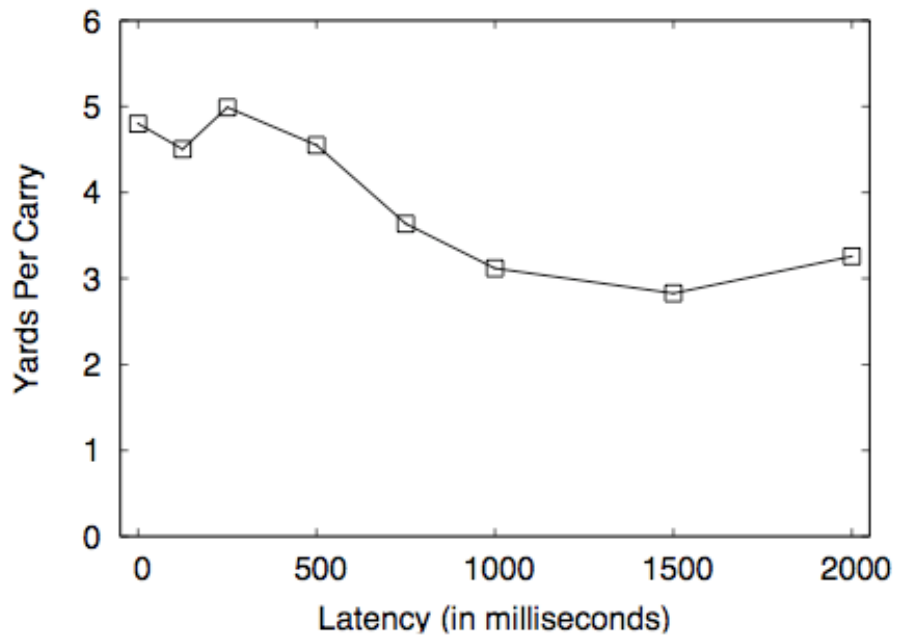
## ■ FPS



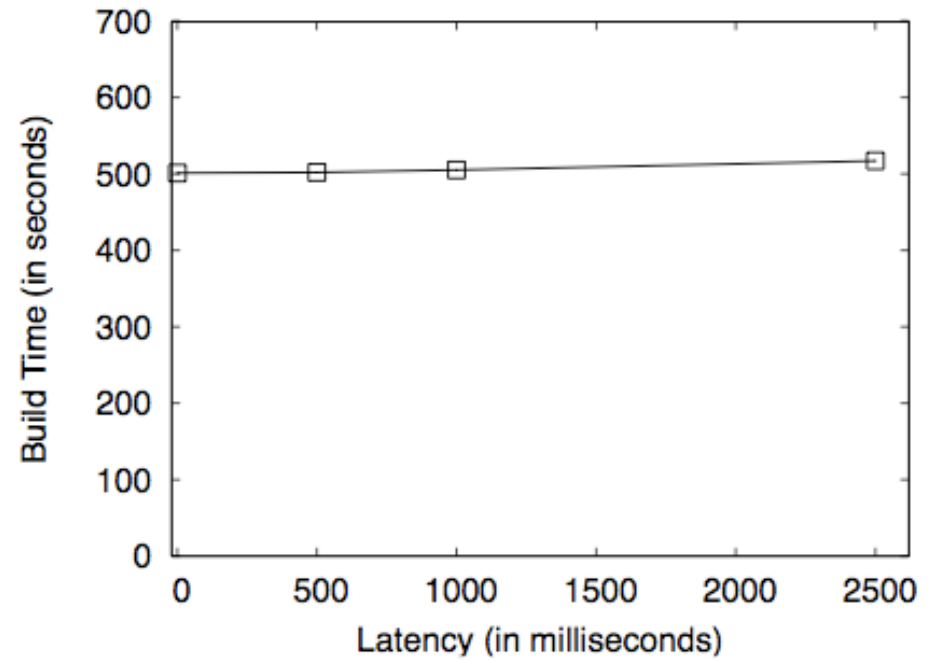
## ■ Racing



## ■ サッカーゲーム

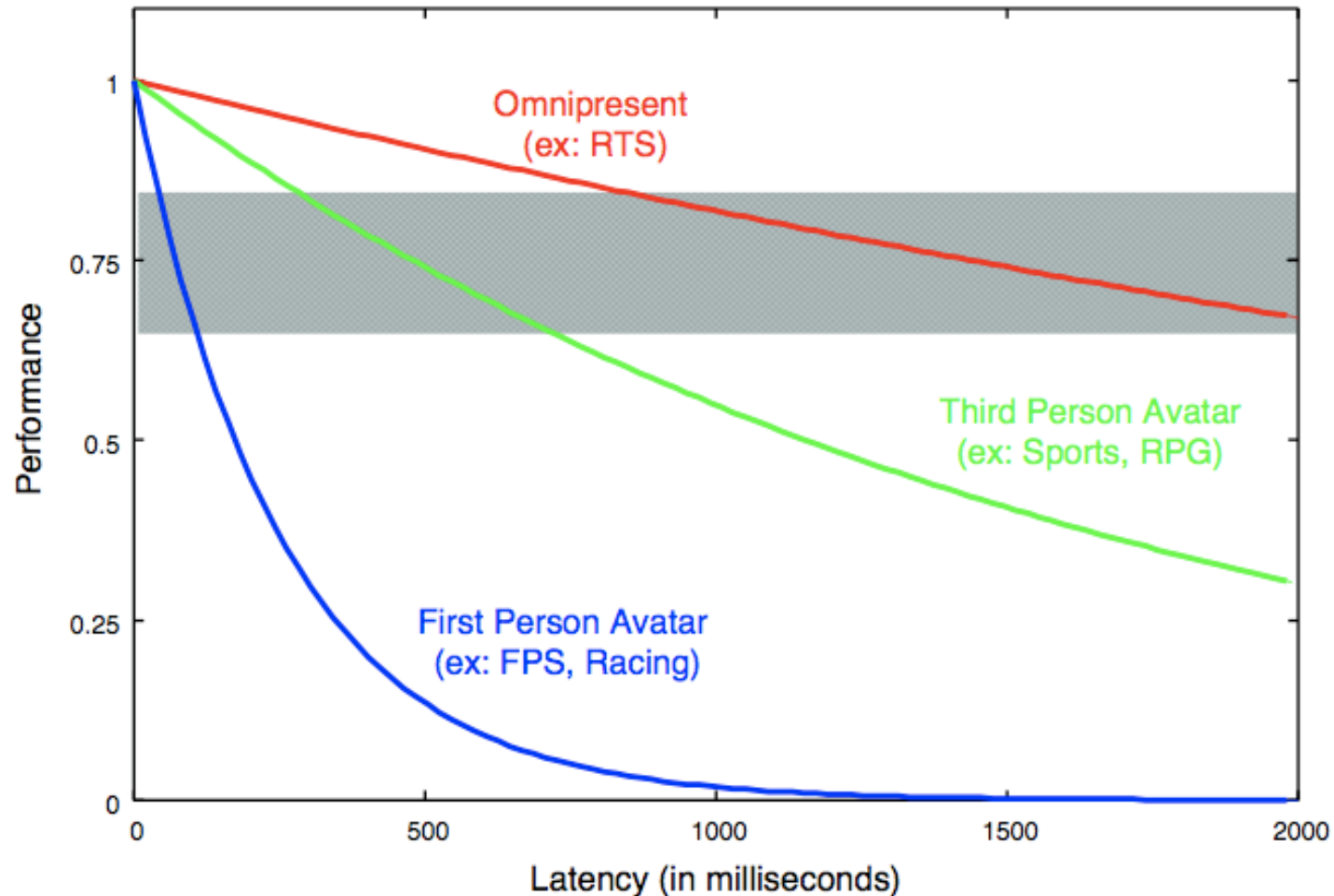


## ■ 戦略ゲーム



# ユーザの遅延に対する反応

- たいていのユーザは50msから300msの遅延を経験



# Networked Virtual Environments

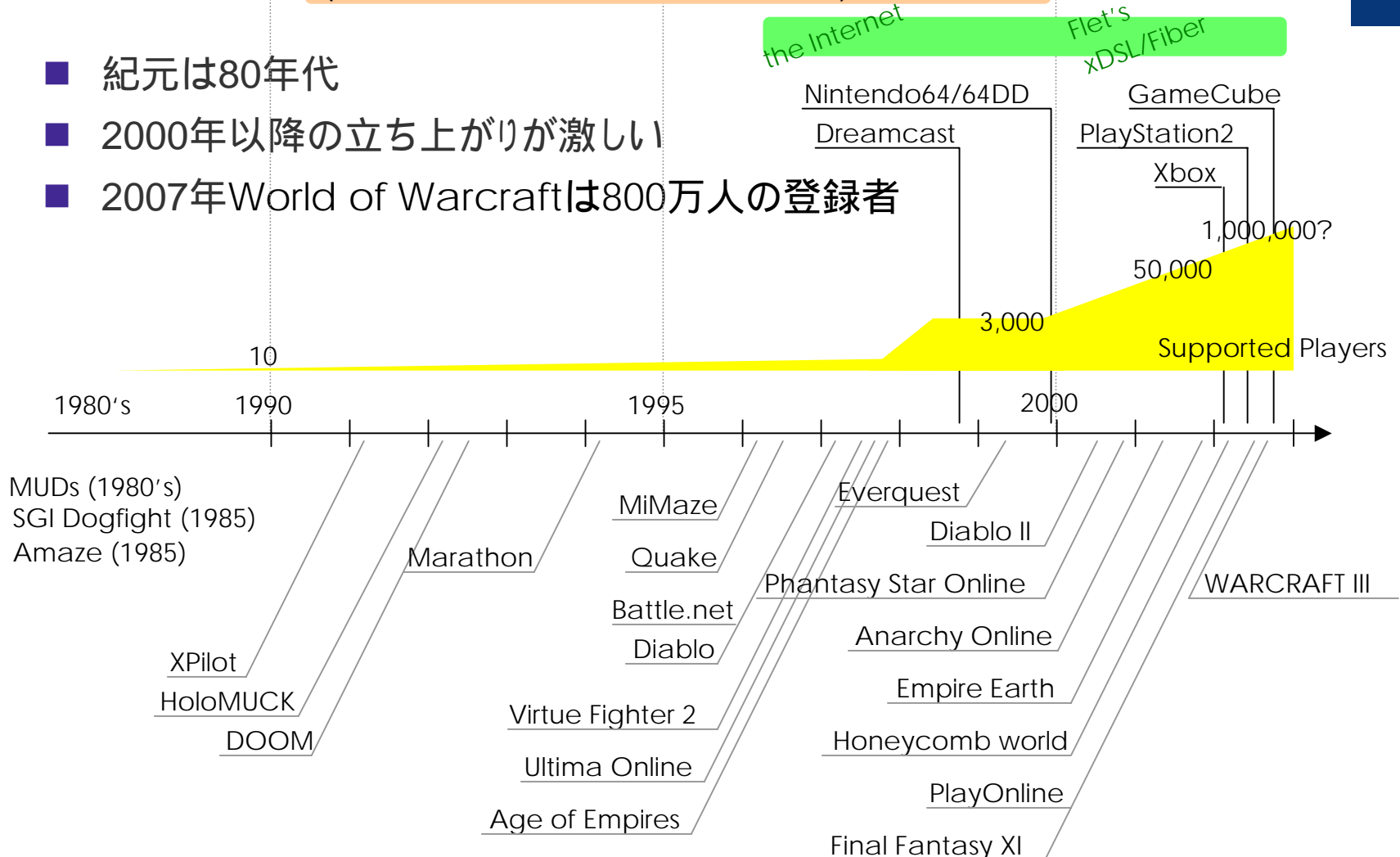
## 歴史とユーザ数の増加

(Military Simulations)

(Collaborative Virtual Environments)

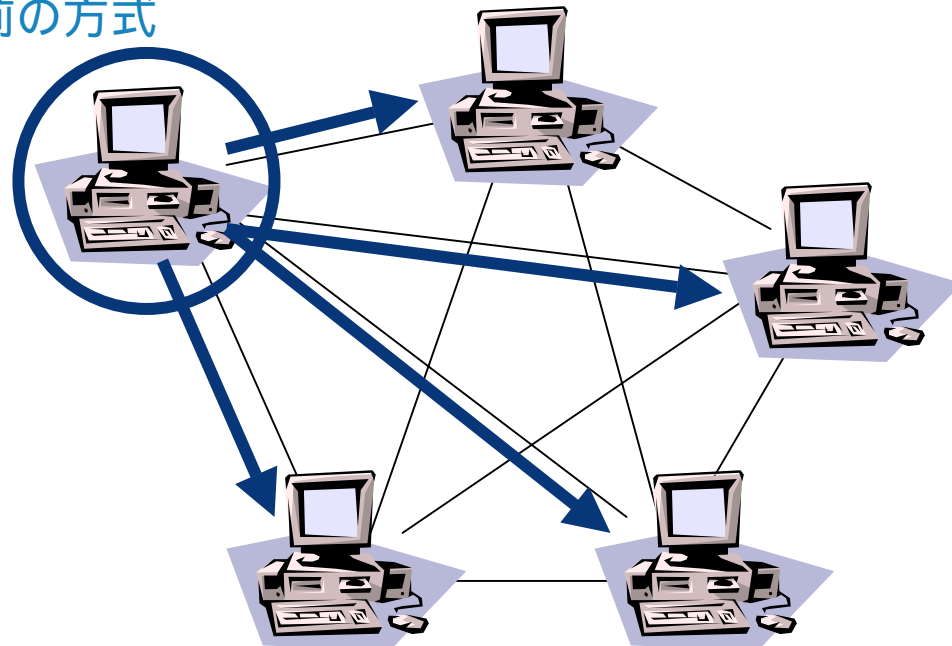
(Online Gaming)

- 紀元は80年代
- 2000年以降の立ち上がりが激しい
- 2007年World of Warcraftは800万人の登録者



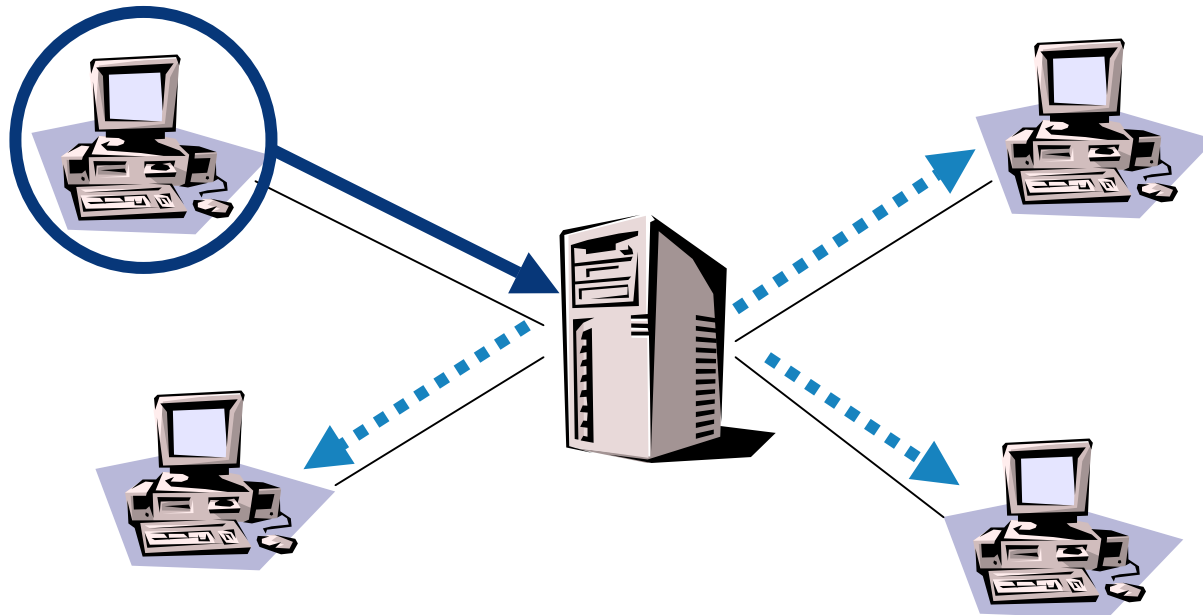
# ネットワークトポロジ(1)

- クライアント主導の複製
  - ユーザがそれぞれにデータベースを保持
  - 更新通知が必要な相手に通知
  - 2000年以前の方式



# ネットワークトポロジ(2)

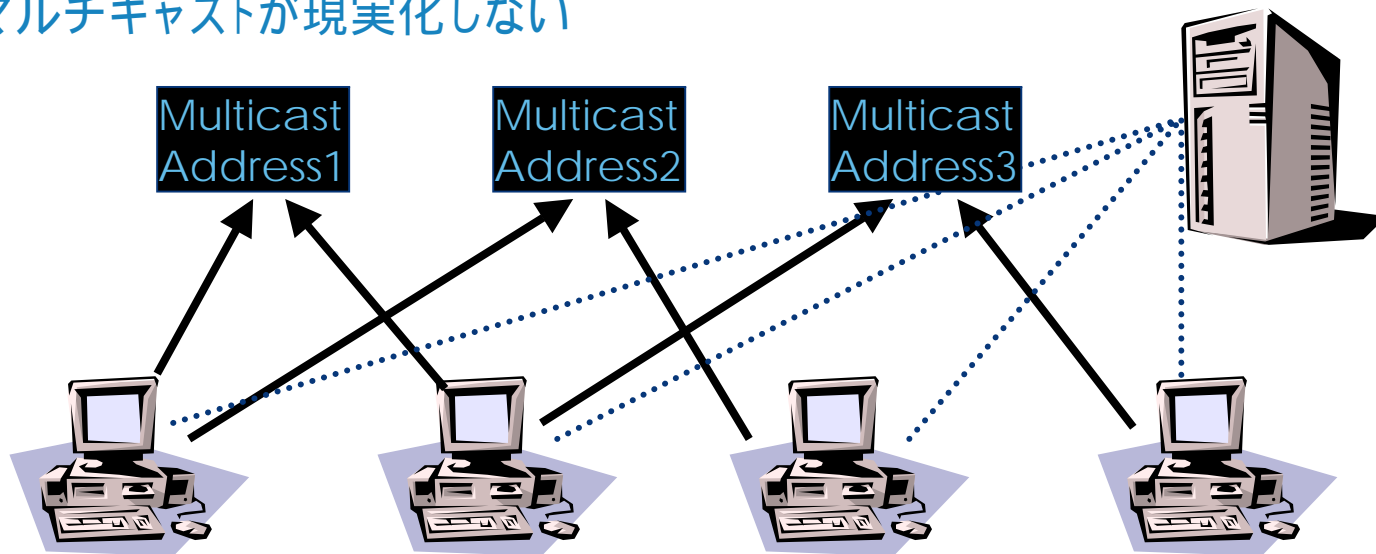
- サーバ集中型
  - データをサーバに送る
  - 現在最も主流な方式
  - 負荷の集中, 維持コスト高

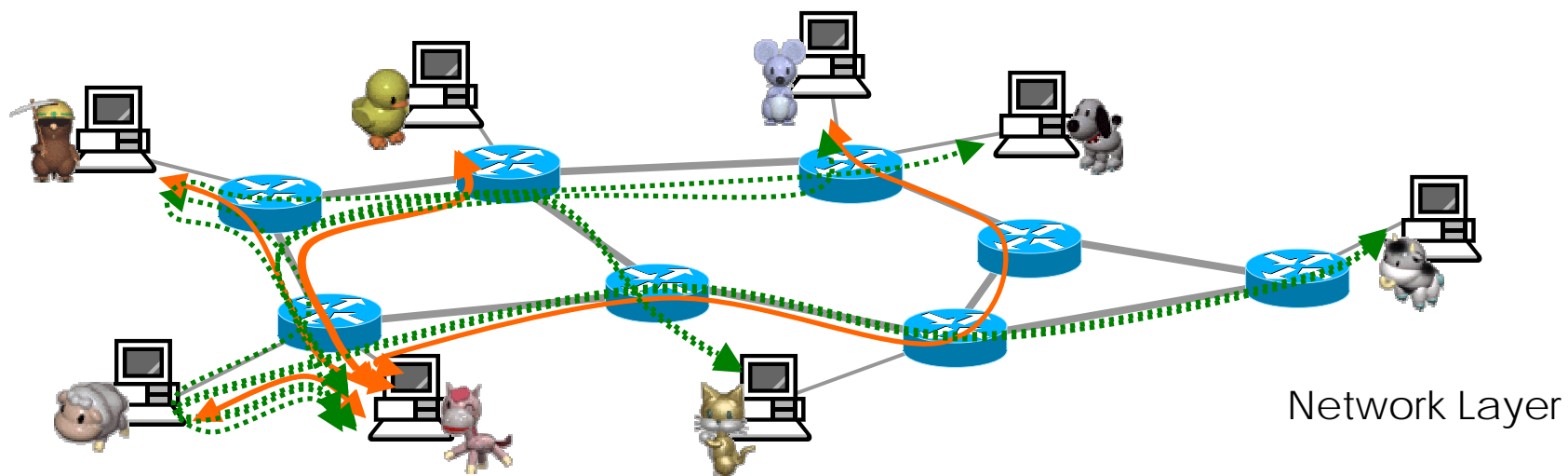
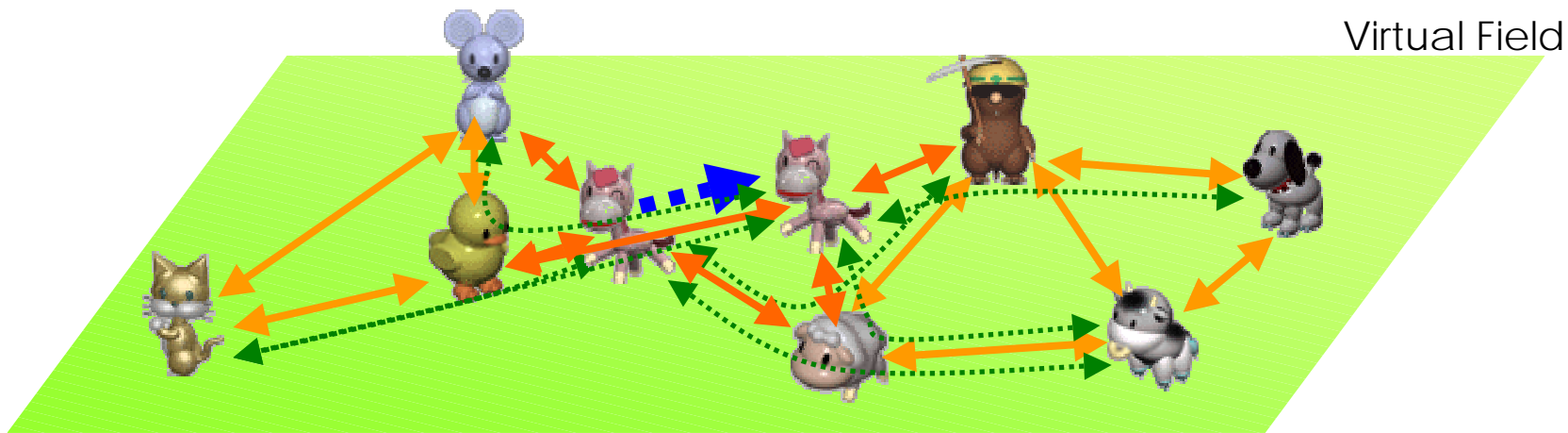




# ネットワークトポロジ(3)

- マルチキャストを用いたグルーピング
  - ゲームの世界をいくつかのエリアに分割しIPマルチキャストを用いてエリア内で通信
  - 90年後半のアイディア
  - マルチキャストが現実化しない





十分に高性能なクライアントマシンのCPU, Storage, Networkを拝借

# P2Pネットワークの構成手法

- オリジナルなP2Pを構成する手法
  - SCAMPI・・・Area of Interestに応じてトポロジを生成
  - 低遅延
- DHTを使う方法
  - Distributed Hash Table
  - 高い耐障害性と一貫性
- SKYPEのP2Pネットワークを利用する方法

# Scampi



自分がインタラクション中のエンティティの数

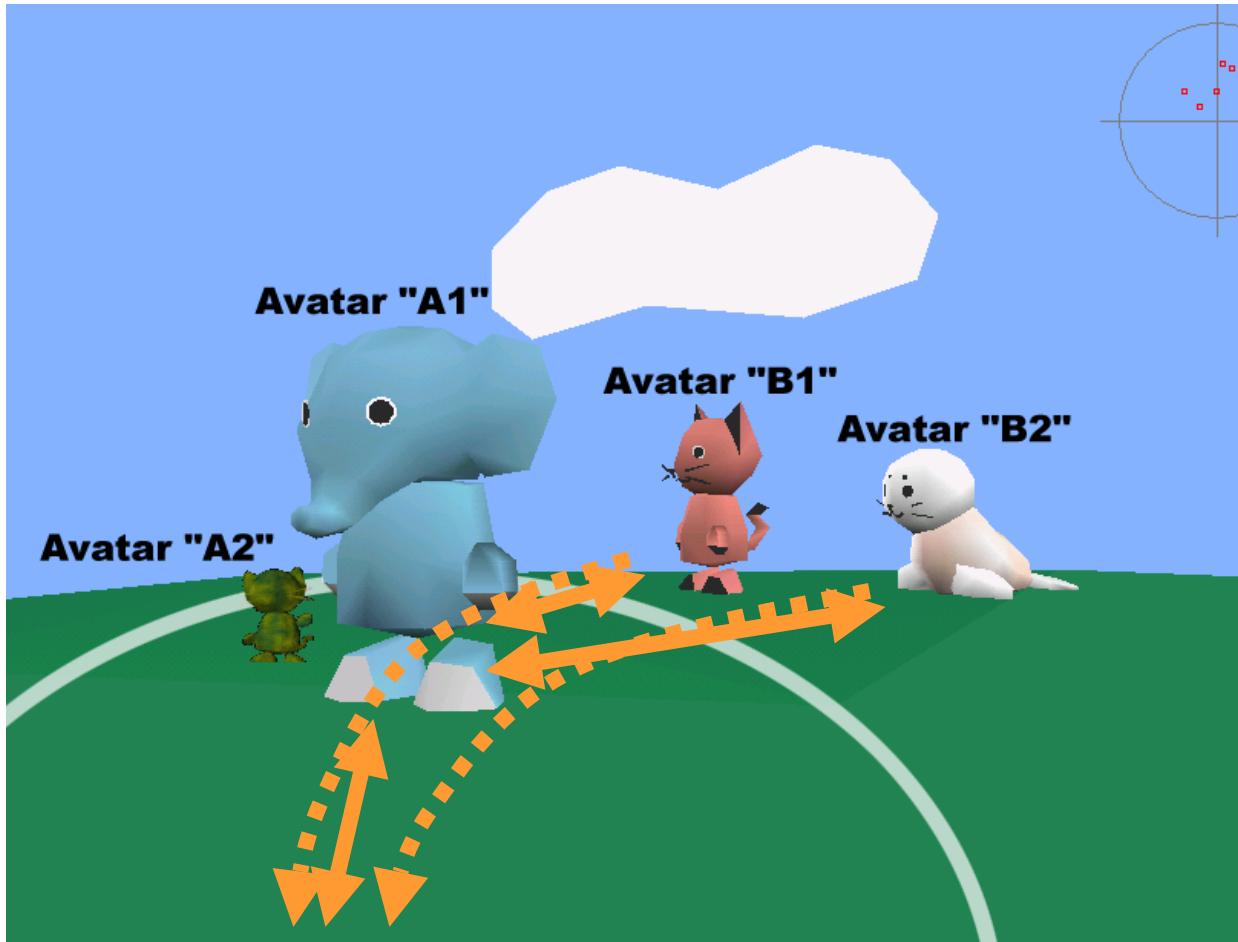
全エンティティの数

- 通信の局所性:すべての参加者とインタラクションする必要は無い
- ユーザ同士の「興味」「関係性」に応じたネットワークトポロジの構成

Y. Kawahara, H. Morikawa, and T. Aoyama

" A Peer-to-Peer Message Exchange Scheme for Large-Scale Networked Virtual Environments " Telecommunication Systems, vol. 25, no. 3-4, pp.353-370, March-April 2004.

# 提案手法の概略



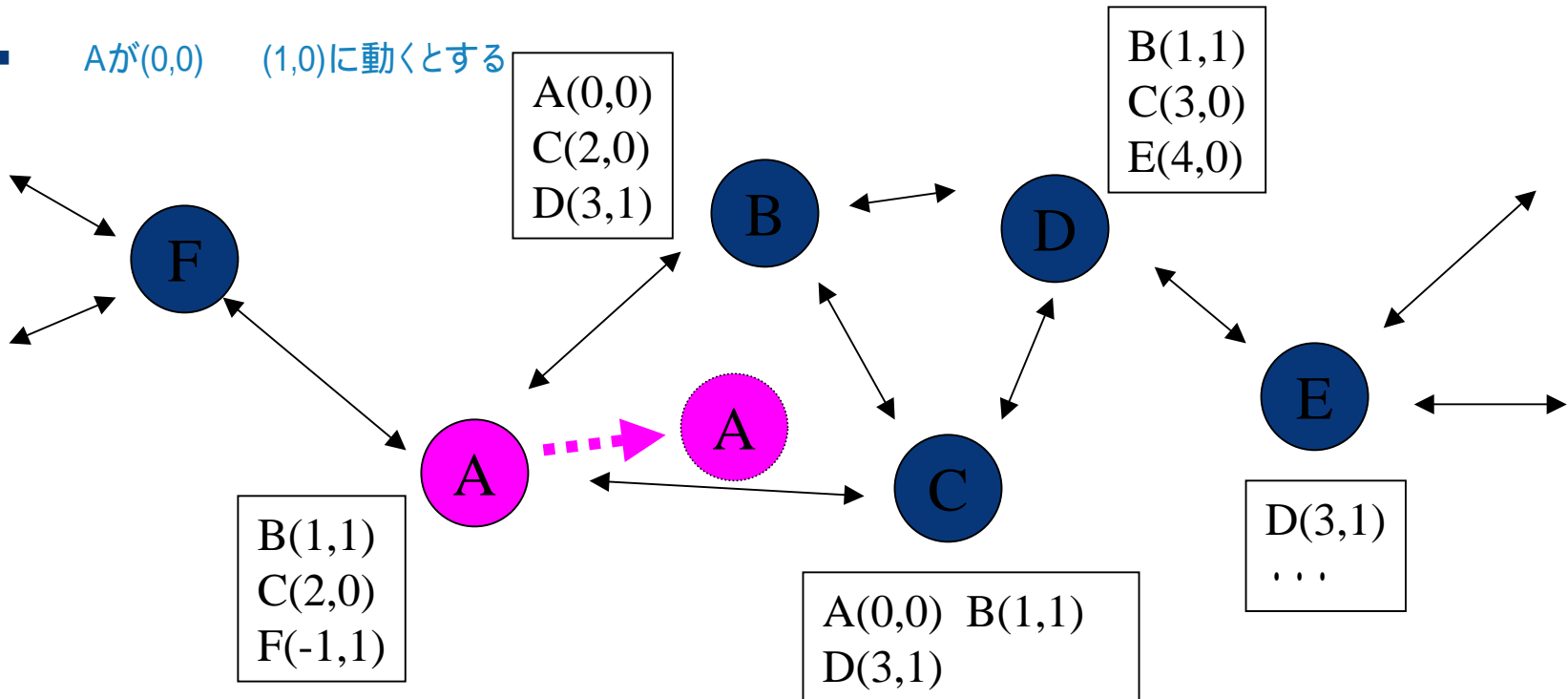
- 仮想空間中で近いエンティティの情報は重要度を高く、遠くなるほど低く
- 詳細な更新データを必要とするのは隣接ノードのみ
- 仮想空間中で離れた場所に位置するエンティティの情報は隣接ノードが転送

# 前提条件

- Net-VEでは以下のことが前提
  - 参加者はあらかじめ**共通の仮想空間の地図**を持つ
  - 参加者は、**仮想空間を自律的に動き回る**
  - 仮想空間上の位置は各エンティティの**自己申告**に基づく

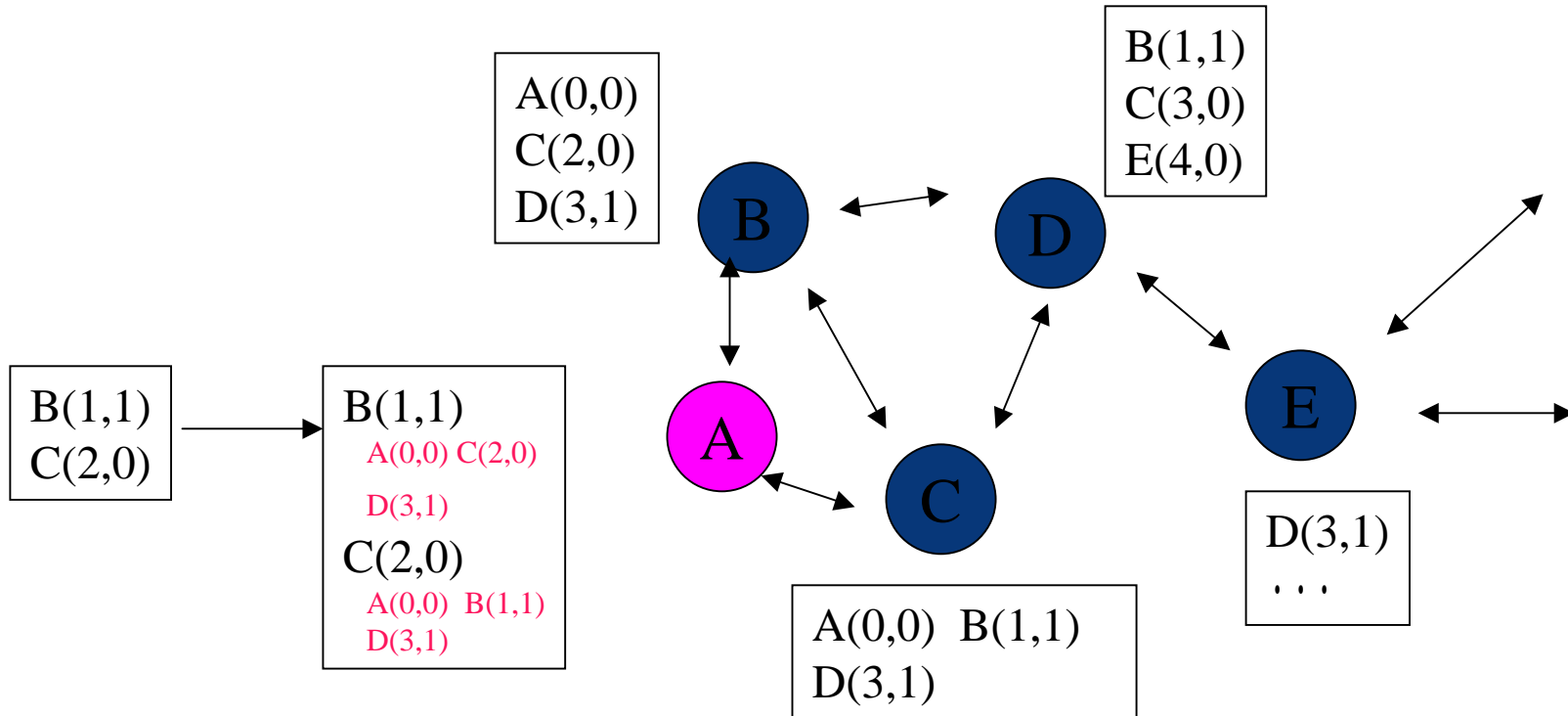
# 位置情報の交換 (1/3)

- システムの開始時には隣接ノードの情報がブートストラップサーバから供給される
- 定常状態では最寄の3エンティティと接続
- 各Entityは、隣接リストを持つ (ノードのネットワーク識別子と位置)
- Aが(0,0) (1,0)に動くとする



# 位置情報の交換(2/3)

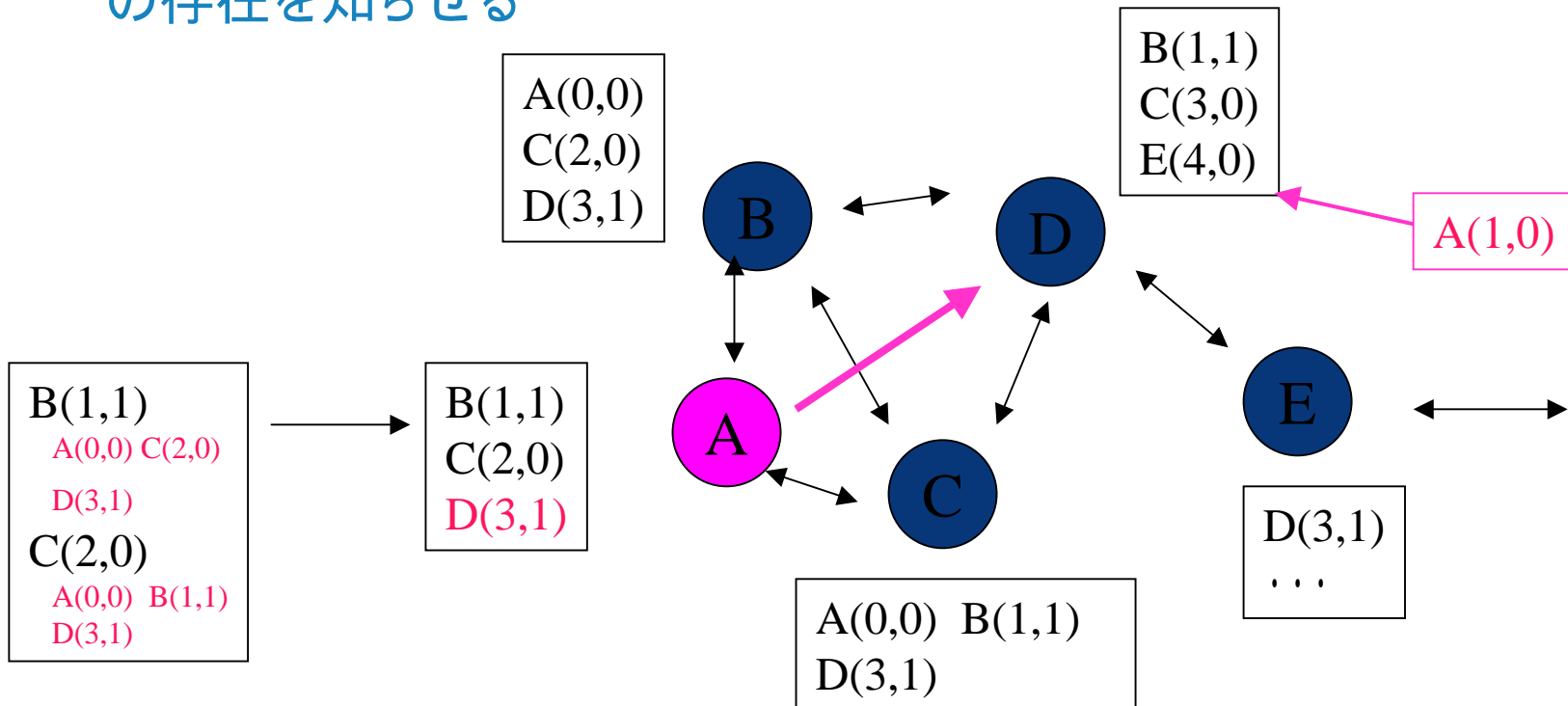
- 隣接エンティティに接続して、位置とidを取得
- さらに隣接エンティティの位置とIDも取得





# 位置情報の交換(3/3)

- 整理して、近い順に3人を新しい隣接エンティティとしてセット
- さらに新しい隣接エンティティに追加されたエンティティに対して自分の存在を知らせる



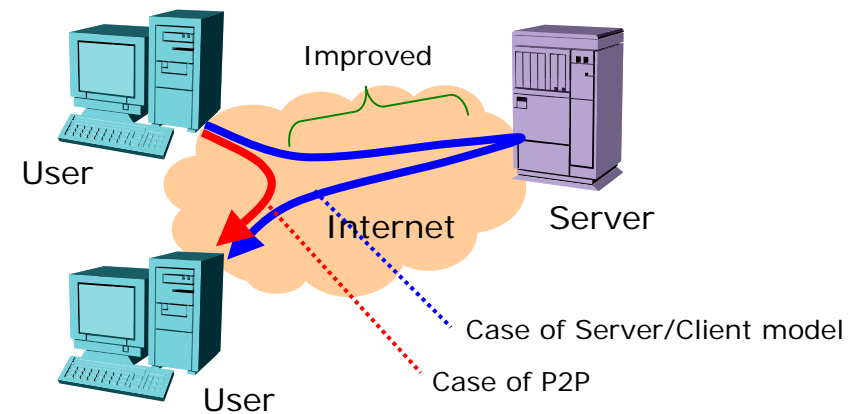
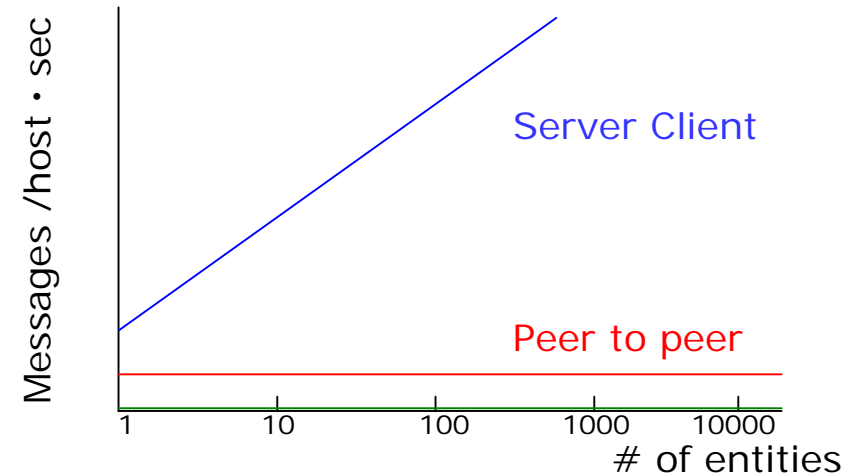
# 特徴

## ■ メリット

- No bottleneck
- Low cost
- Low latency

## ■ デメリット

- ユーザ間のデータ一貫性の維持
- 仮想空間中で「ワープ」ができない

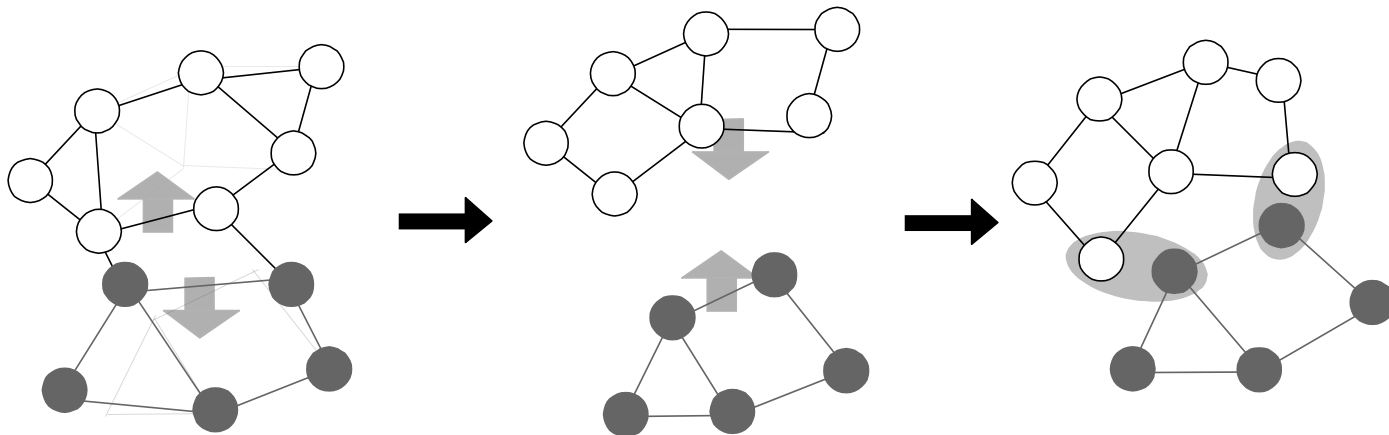


# 問題点:P2Pネットワークの分断

## ■ ピアツーピアネットワークの分断問題

- 直近のNエンティティと通信し続けていると, 仮想空間中でのエンティティの地理的な局在性が原因でN+1以上のエンティティからなる分断ネットワークが発生
- 一度発生した分断ネットワークは, 距離が近づいても再接続できない

互いにリアルタイムに情報を把握すべきエンティティ間で一貫した情報共有が行われない



# 分断修復のアプローチ

## ■ 要求事項

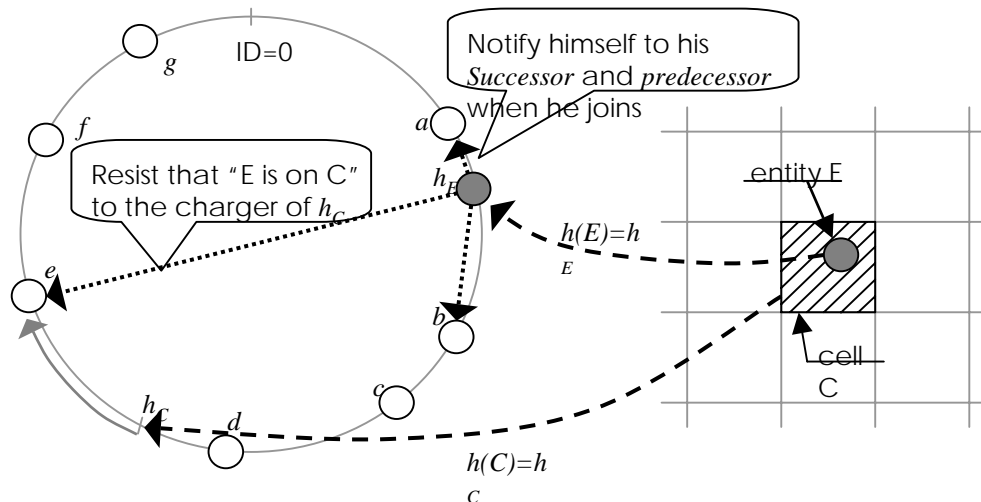
- スケーラブル・低負荷
  - ピアツーピアを用いた更新情報共有手法の効果を打ち消さないため
- 分断に関する情報を用いない
  - このネットワーク分断は検出不能であり、いつ誰が分断ネットワークを形成したかといった情報が得られないため

## ■ アプローチ

- 「マクロな視点」の位置管理にDHTの利用 (たとえばChord)
- 「ミクロな視点」の位置情報交換はこれまで通り
- エンティティと位置の対応をソフトステートに管理
  - 位置情報を管理するためのオーバーレイネットワークを構成

# 分断ネットワークの修復手法 (1/2)

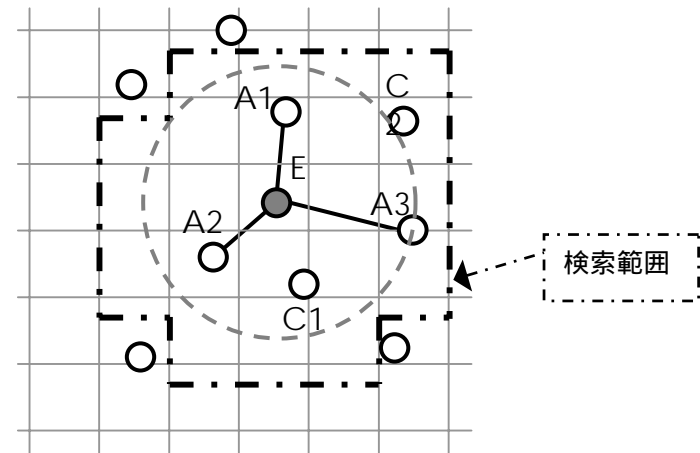
- 準備
  - Chordトポロジーの構築と維持
    - エンティティIDや通信識別子をハッシュにかけてノードIDとする
- 位置登録処理
  - 各エンティティは定期的に「自分がどのセルに存在しているか」を登録
    1. セルIDをキーとしてChord上での担当ノードを取得
    2. 自エンティティがそのセルに存在していることを担当ノードに登録



# 分断ネットワークの修復手法(2/2)

- 近隣エンティティの検索処理
  - 各エンティティは定期的に自分の近隣にいるエンティティを検索
    1. 検索対象セルの判定
      - 全てのAEが含まれる範囲 or 自分が情報を必要とする範囲 に相当するセルを対象とする
    2. AE候補エンティティの取得
      - 各セルの担当ノードに、現在そのセルに属しているエンティティのリストを知らせてもらう
    3. エンティティの現在座標の取得
    4. AEリストの更新

「全てのAEが含まれる範囲」を検索対象とする場合の例



その他の処理は、原則的にオリジナルのChordに従う

- トポロジーの維持のためのメッセージング
- 耐障害・効率化のための冗長化
- etc...

# 特徴と応用

## ■ 特徴

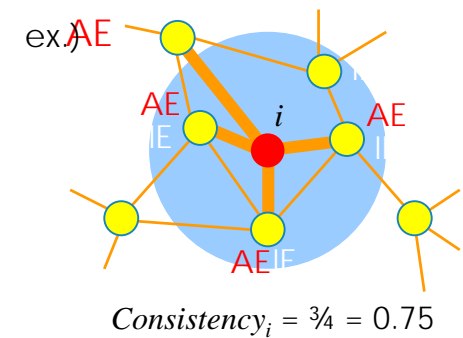
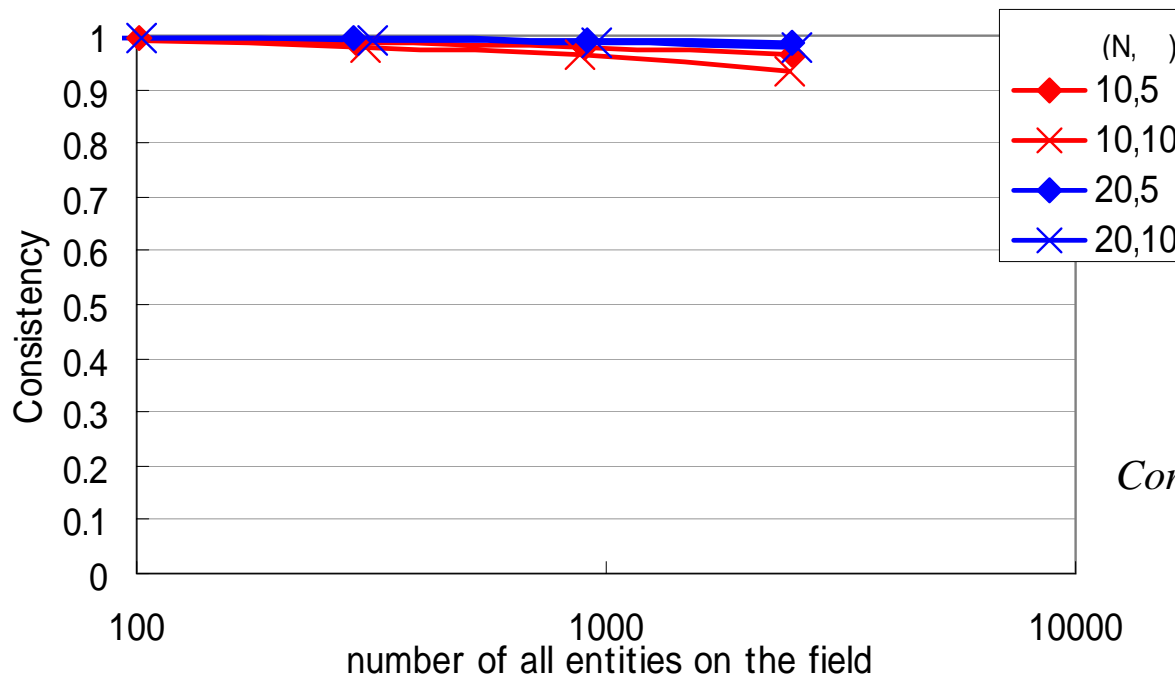
- スケーラブル
- 低コスト
  - 本手法を用いることによる負荷の増加は...
    - Chordトポロジーの維持のためのStabilization (定期的に両隣接ノードと通信)
    - 新規参加通知 (両隣接ノード及び finger table に影響がある範囲に送信)
    - 退出通知 (同上)
    - 位置登録処理 (定期的に担当ノードに送信)
    - 近隣エンティティ検索処理 (定期的に各セルの担当ノードに送信)  
それぞれ通信対象と頻度が限られている

## ■ 応用

- 高速移動・瞬間移動のサポート
  - 情報共有の範囲がAE間のみには制約されず, 位置に応じたエンティティ発見が可能
- アイテムなど, ユーザエンティティに属さないオブジェクトの管理

## シミュレーションによる一貫性の評価

信頼性のないP2P Overlayだが, C/Sと遜色ない一貫性を保持可能  
直近のエンティティ同士はUnicastで結ばれるため低遅延



$$Consistency_i = \frac{n(AE \cap IE)}{n(IE)}$$

- $n(X)$  : The number of a set  $X$
- $AE$  : The set of Active Entities
- $IE$  : The set of the nearest  $n(AE)$  entities



# SCAMPI

- SCAAble Multiuser community Place on the Internet
- 3D chat application software using Peer-to-Peer technology

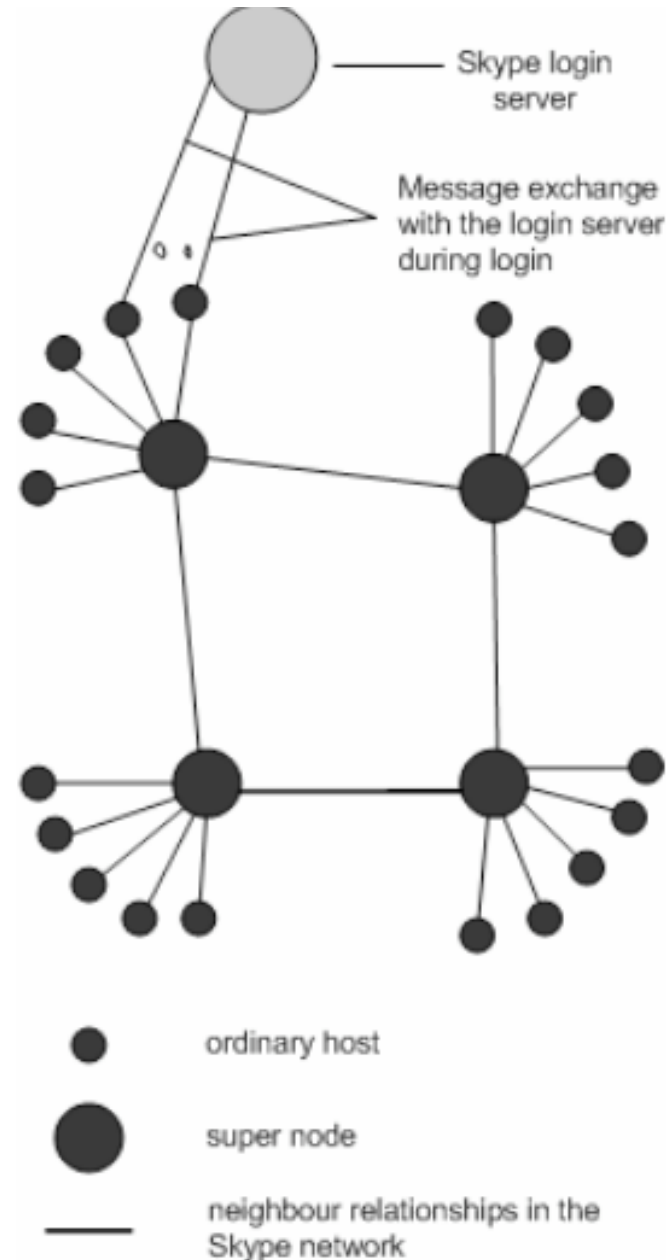


# Skype 4 games(1)

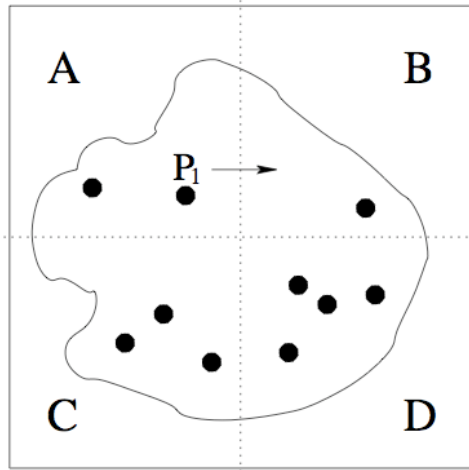
- Mannheim 大学 Tonio Triebelらによる研究
  - NetGames 2007
- 目的
  - P2Pのプロトコルやインフラをスクラッチから用意するのはそれなりに大変
  - 「ありものの」の技術を使って低コストでゲームを作れないか
- 手段
  - Skype のAPIを利用する

# Skype

- P2P技術を利用したIP通話サービス
  - 電話帳以外にサーバを持たない
  - Voiceのほかvideo, text chatもサポート
  - APIを公開している
  
- An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol
  - Salman A. Baset and Henning Schulzrinne



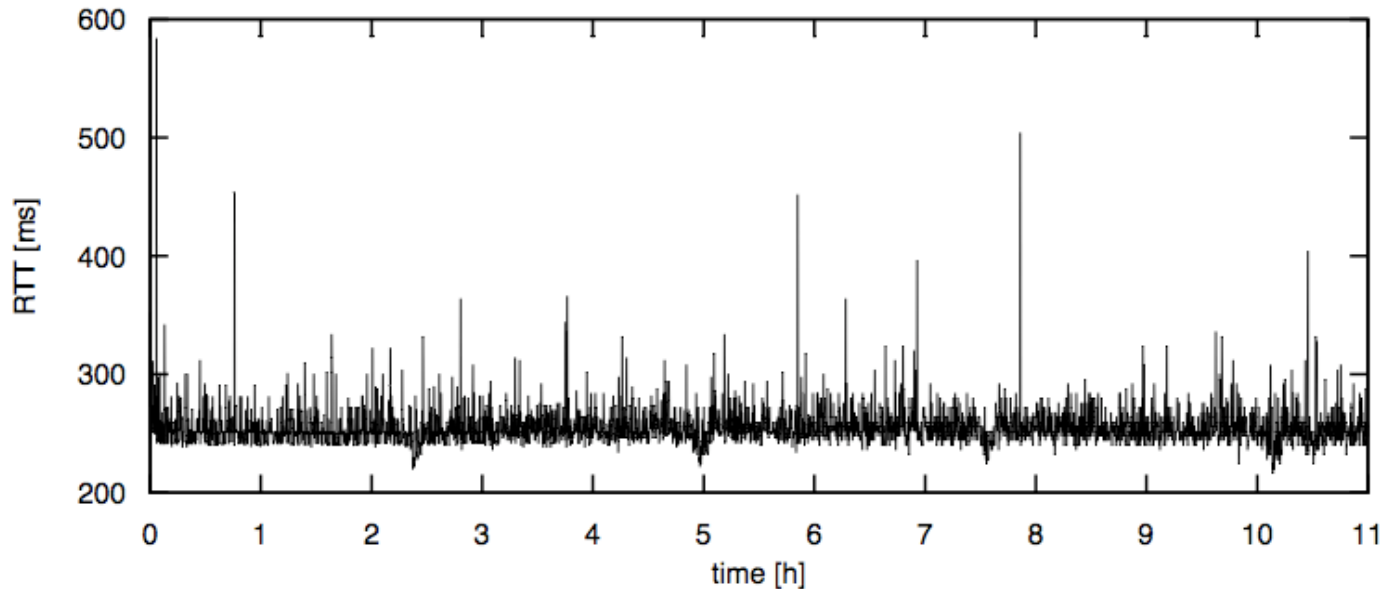
# Skypeの利用



エリアを分割し, それぞれのエリアにSkypeの public chatのチャンネルを利用.

ユーザはエリアに対応するPublicChatに参加する

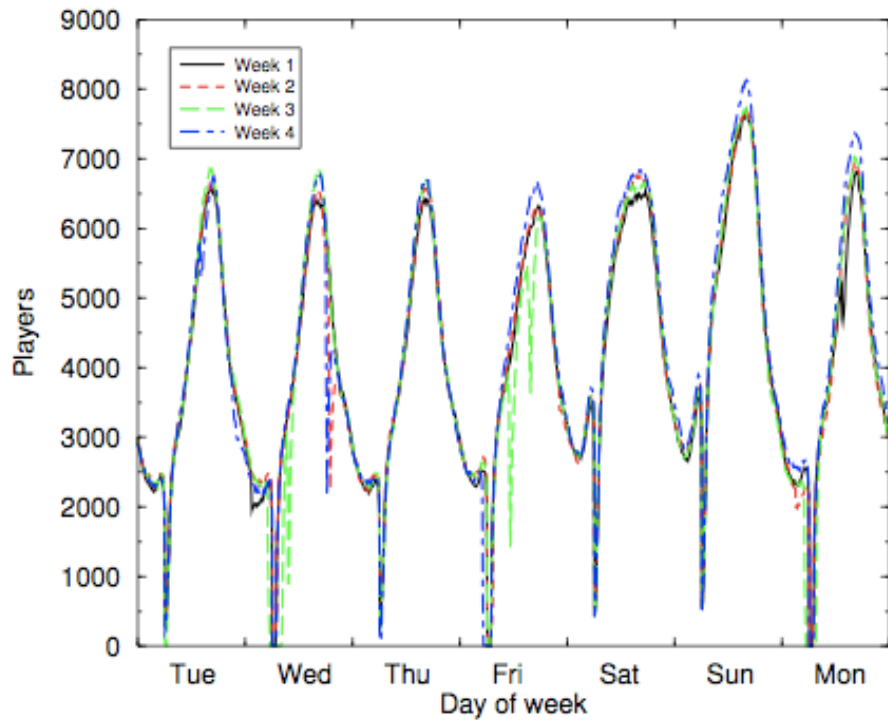
性能も上々



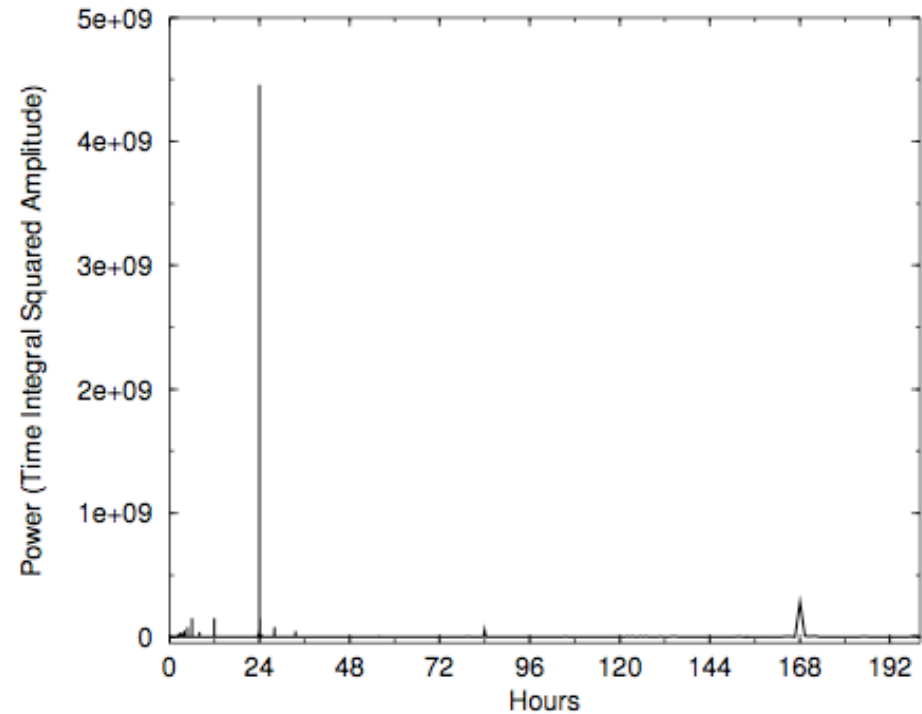
ドイツー  
カナダ間の  
往復遅延

# ユーザの利用パターンに関する調査研究

- A long term study of a popular MMORPG
  - Wu-chang Fengほか, Netgames 2007
  - EVE Onlineという実在のサービスのログを解析
- スケールするようにシステムを設計することは大切だが, 品質を維持しながら運営することはもっと大切



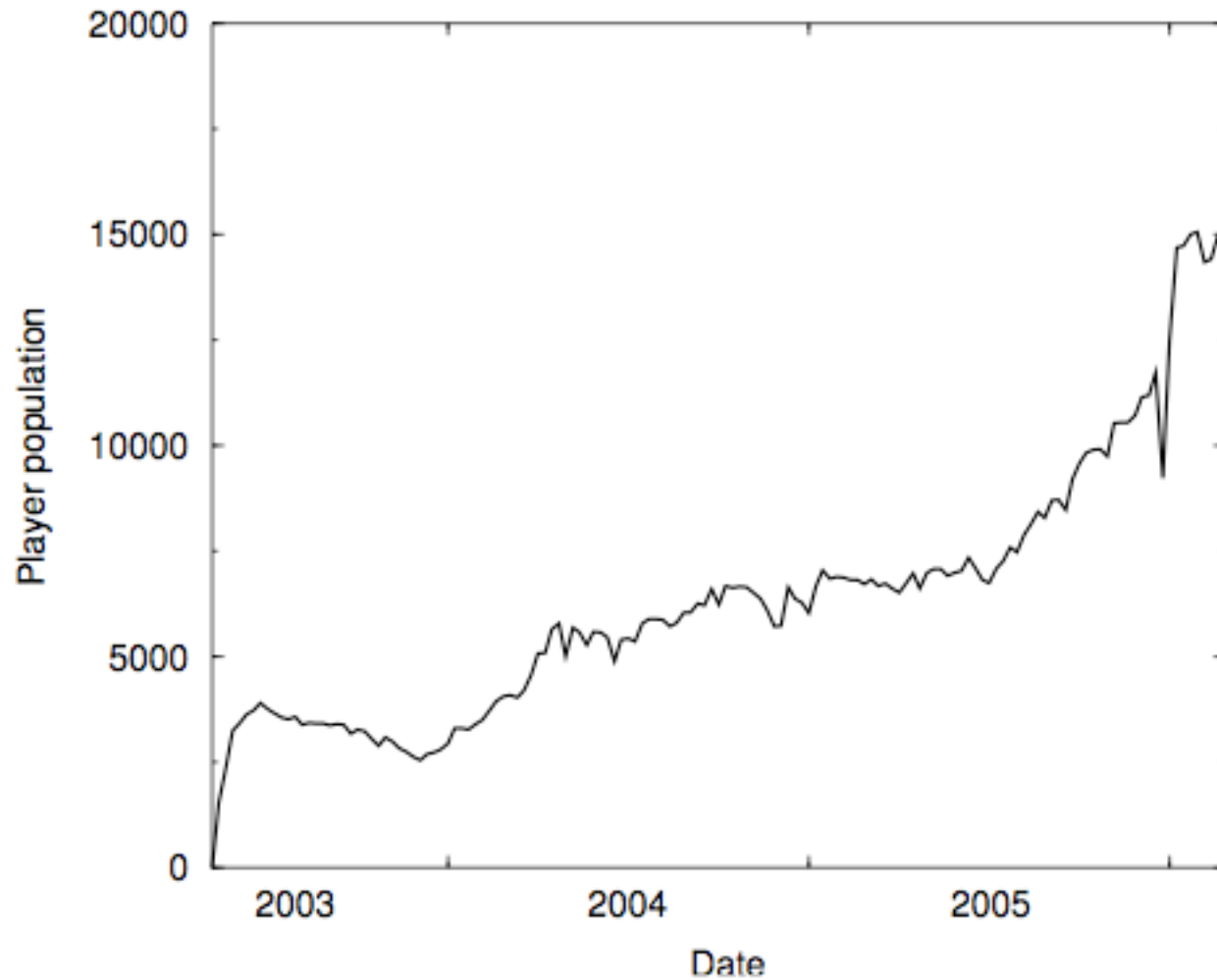
(a) Weekly player load (March 2004)



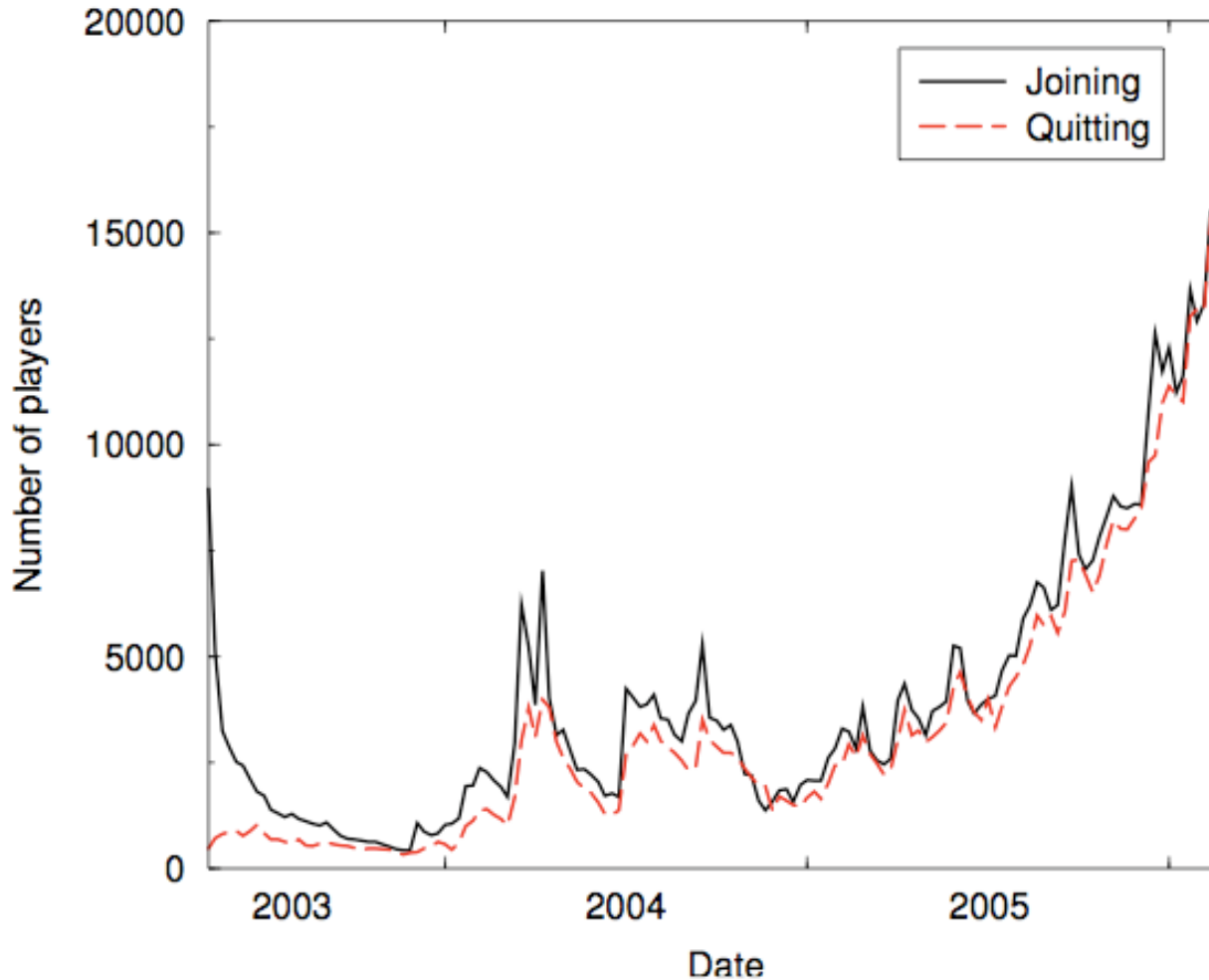
(b) FFT of player load

24時間ごとと, 7日ごとに周期性

# プレイヤーの増加は読めない



# 加入と退会の割合

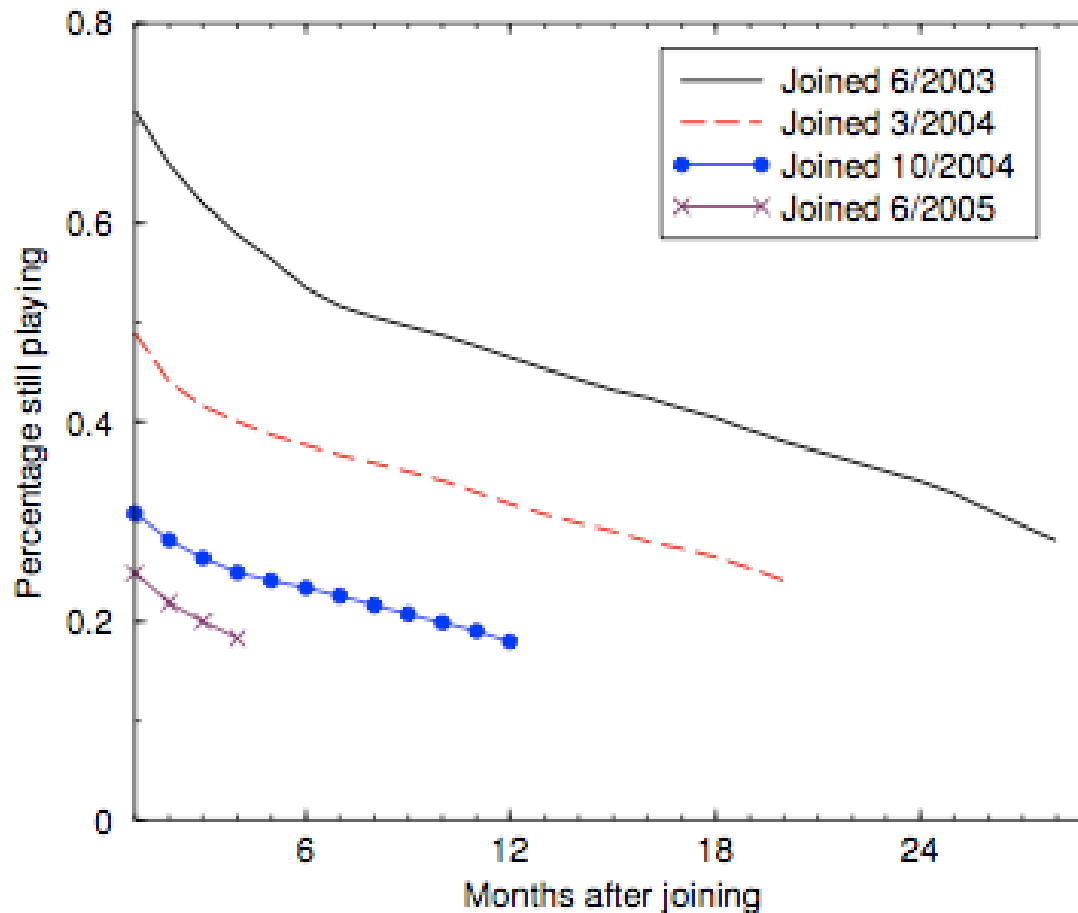


加入と退会の  
相関が高いのは、  
「お試し」ユーザ  
が多いため。

一時的な盛り上がりは  
プロモーション活動  
によるもの

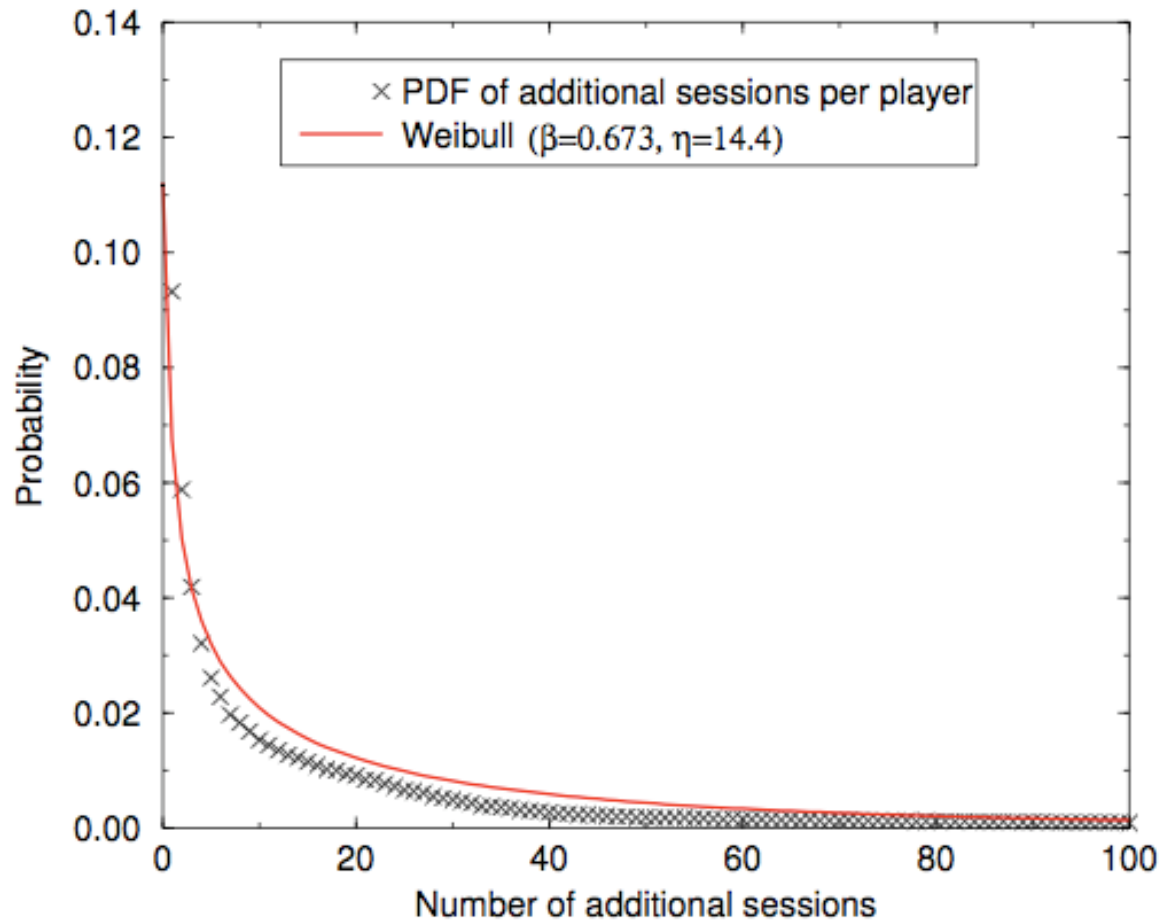


# 参加のタイミングと退会の関係



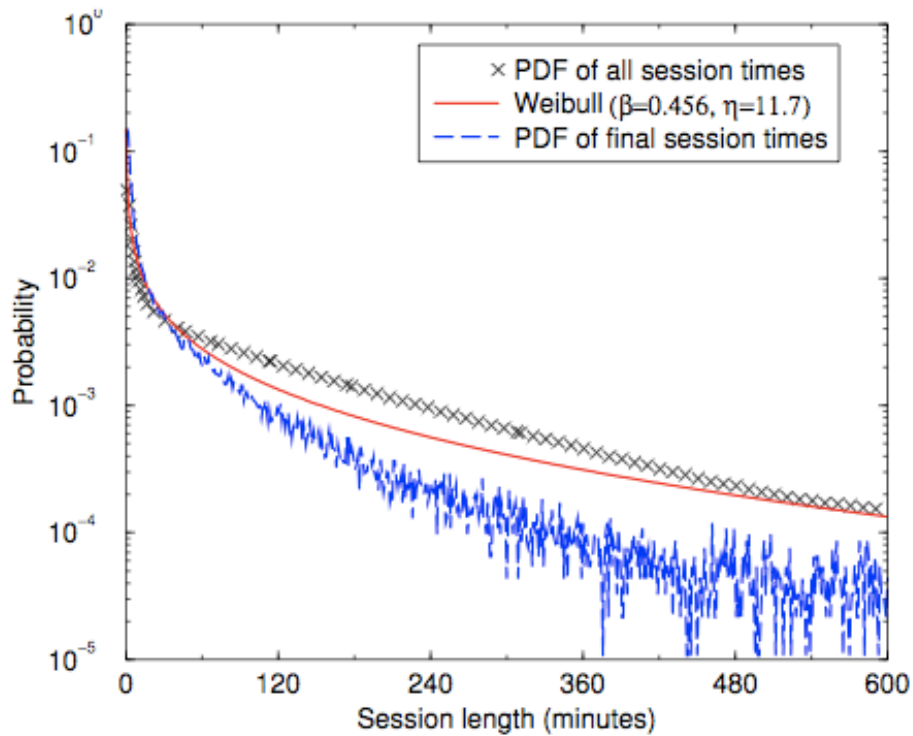
古参ほど長くとどまる  
新参加者はすぐ退会する

# 「去り行くユーザ」の検出

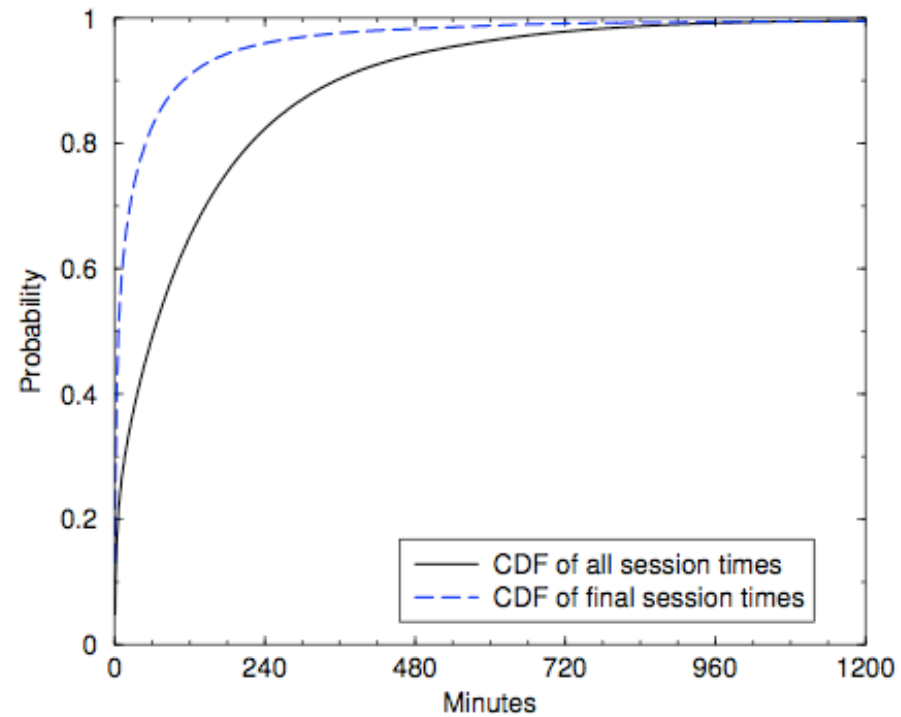


ほとんどのユーザが  
2, 3回ログインして  
やめて行く

# やめる直前のセッションは短い！

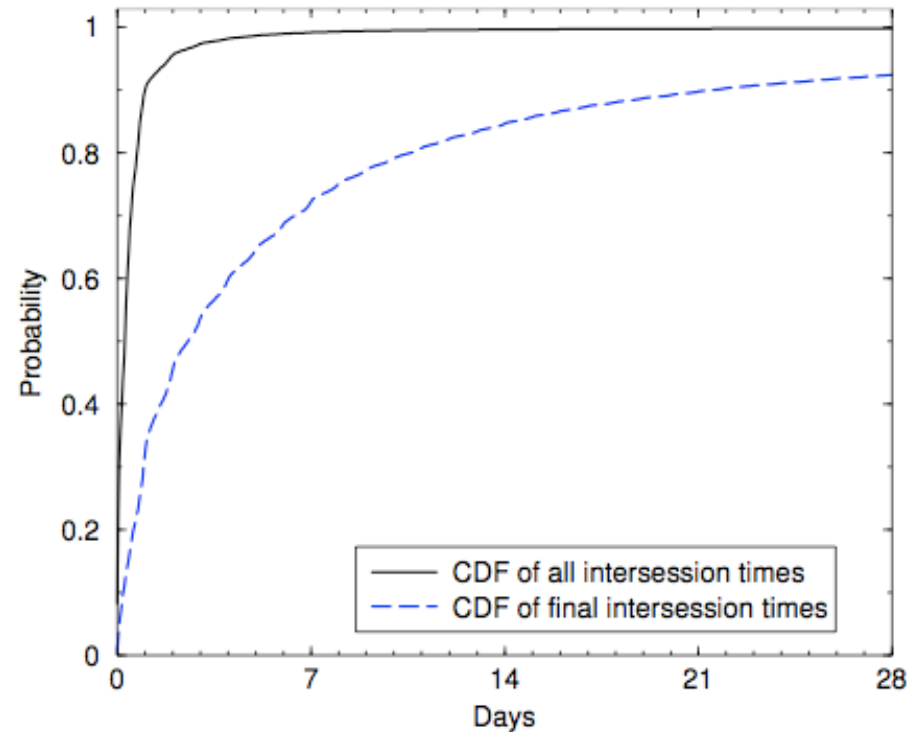
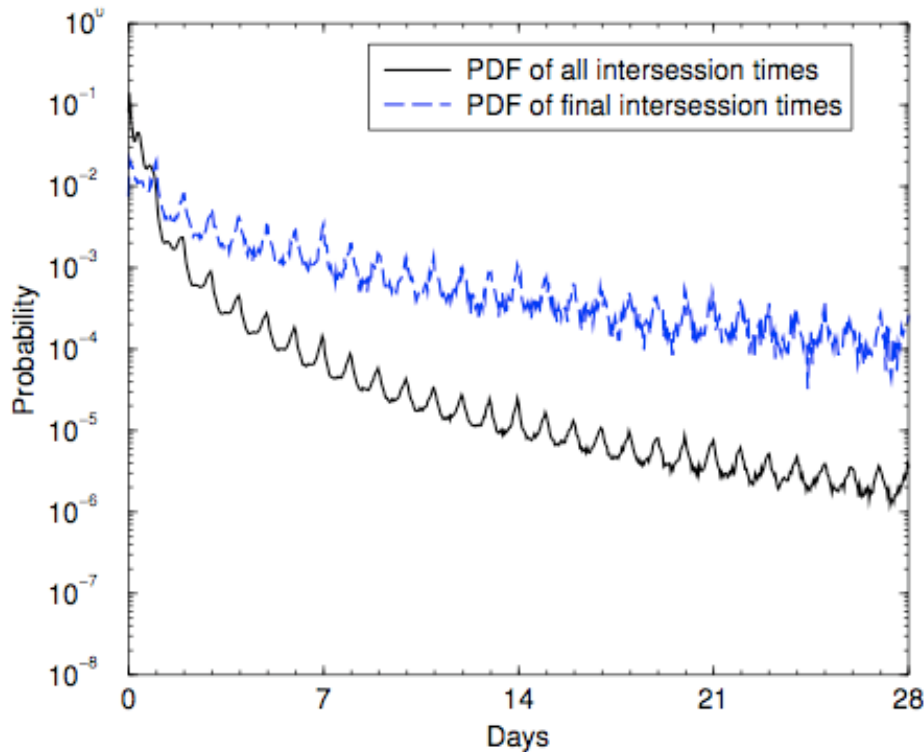


(a) PDF



(b) CDF

# やめる決心をするまでの時間間隔



やめそうなユーザを検出するには、「最近ログインしていなかったがちょっとだけログインしてみた」ユーザを捕まえることができれば...

Ubiquitous  
Computingへの  
展開

# ゲームにとってのユビキタス技術

## 端末技術

- DS, PSP,携帯電話などの端末技術はだいぶ整った

## 無線通信

- 無線通信はバッテリー食い
- エラーが多い

## センサ技術

- 位置情報がとれる環境は整ってきた
- 加速度センサもそれなりに成熟
- 遊べるセンサは何か
  - 温度, 湿度, アルコール?

## (表示技術)

- Augmented realityはやっぱり憧れ?

# ゲーム+ユビキタスコンピューティング

- 実空間と仮想空間の密な連携
  - ゲームフィールドとしての実空間の利用
    - 実空間を移動してプレイする
      - コンソールからの解放
  - パラメータとしての実空間の利用
    - 時々刻々と変化する実空間の情報をゲームのパラメータとして用いる
      - プログラムされた世界からの解放

まったく新しいエンターテインメントの可能性  
ユビキタスのテストベッドや商用サービスとして注目されつつある

さまざまなテストベッドを試作・評価して、  
ユビキタスコンピューティング環境における  
キラーアプリケーションを模索

## 2つのアプローチ

- 環境中のセンサから取得される実空間情報を利用したアプリケーション
  - ユーザがいなくても、時々刻々変化する実空間情報を元に、アプリケーションが自律的に動作
    - 実空間指向モンスター収集ゲーム
      - ユビキタスモンスター
- ユーザの持つセンサから取得される実空間情報を利用したアプリケーション
  - ユーザごとに異なる情報を元に、コンテキストウェアなアプリケーションを実現
    - 位置情報履歴を利用したモバイルエンターテインメントシステム
      - 例: 実空間マインスイーパ





# 位置情報を活用したサービス

- GPSの低価格化により位置情報を使ったゲームが拡充
  - Mapion ケータイ国盗り合戦
  - au Smart Sports Run&Walk
- 研究的取り組み
  - Human Pacman
    - by Mixed Reality Lab of National University of Singapore
    - 2004



# “ユビキタスモンスター” by 東京大学

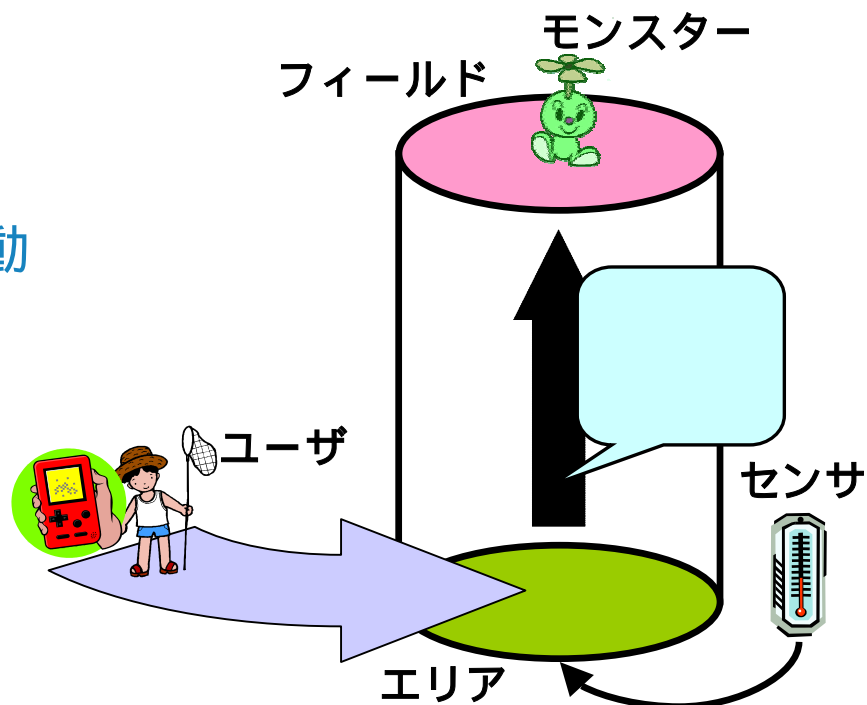
- 実空間情報適応型ネットワークゲーム
  - 従来のモンスター収集ゲーム
    - 例: ポケットモンスター
      - 地域ごとにことなるモンスターの生息する  
仮想空間を旅してモンスターを収集
  - ユビキタスコンピューティング環境へ移植
    - ユーザが実空間を移動してモンスターを収集
    - 時々刻々と変化する実空間情報に応じて, 登場するモンスターが変化
      - 実空間と仮想空間とが密に融合
    - 実空間中のいたるところで, その時・その場の状況に応じたモンスターが登場

モンスターがどこにでもいる = “ユビキタスモンスター”

環境情報適応型コンピュータエンタテインメントへのセンサネットワークの適用  
川西 直 ほか, 情報処理学会, エンタテインメントコンピューティング, 2006

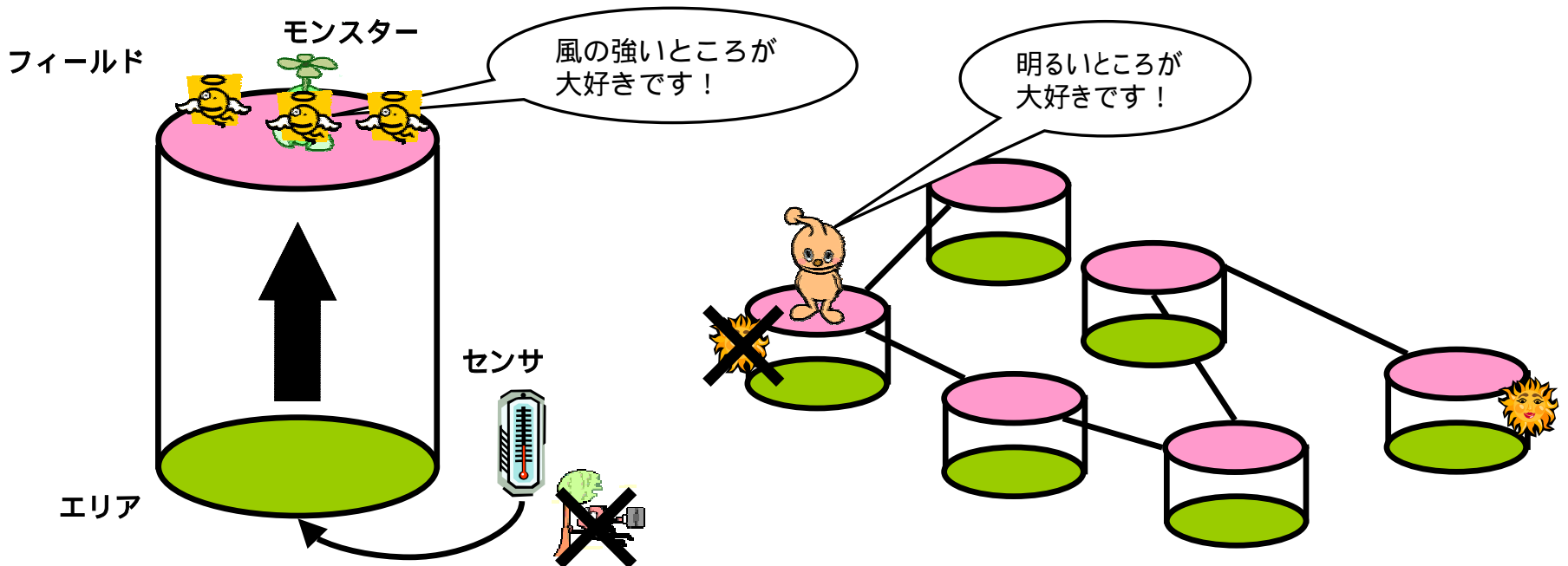
## 基本シナリオ(1)

- 環境に応じたモンスターの生息
  - 実空間中の「エリア」 仮想空間中の「フィールド」
  - センサなどから環境情報を取得
- ユーザによるモンスターの捕獲
  - 実空間中でエリアに接近
  - 対応するフィールドに生息するモンスターがユーザ端末上に移動
  - モンスターを捕まえる



## 基本シナリオ (2)

- モンスターは、それぞれ独自に設定された嗜好と、実空間中に設置されたセンサから取得される環境情報をもとに、生息する
  - 好きな場所: 繁栄
  - 嫌いな場所: 衰退
- モンスターは自分の好きな場所を求めて、フィールドのネットワークを自律的に移動する
  - 生き残るために、そして繁栄するために



# ユビキタスモンスター全体像



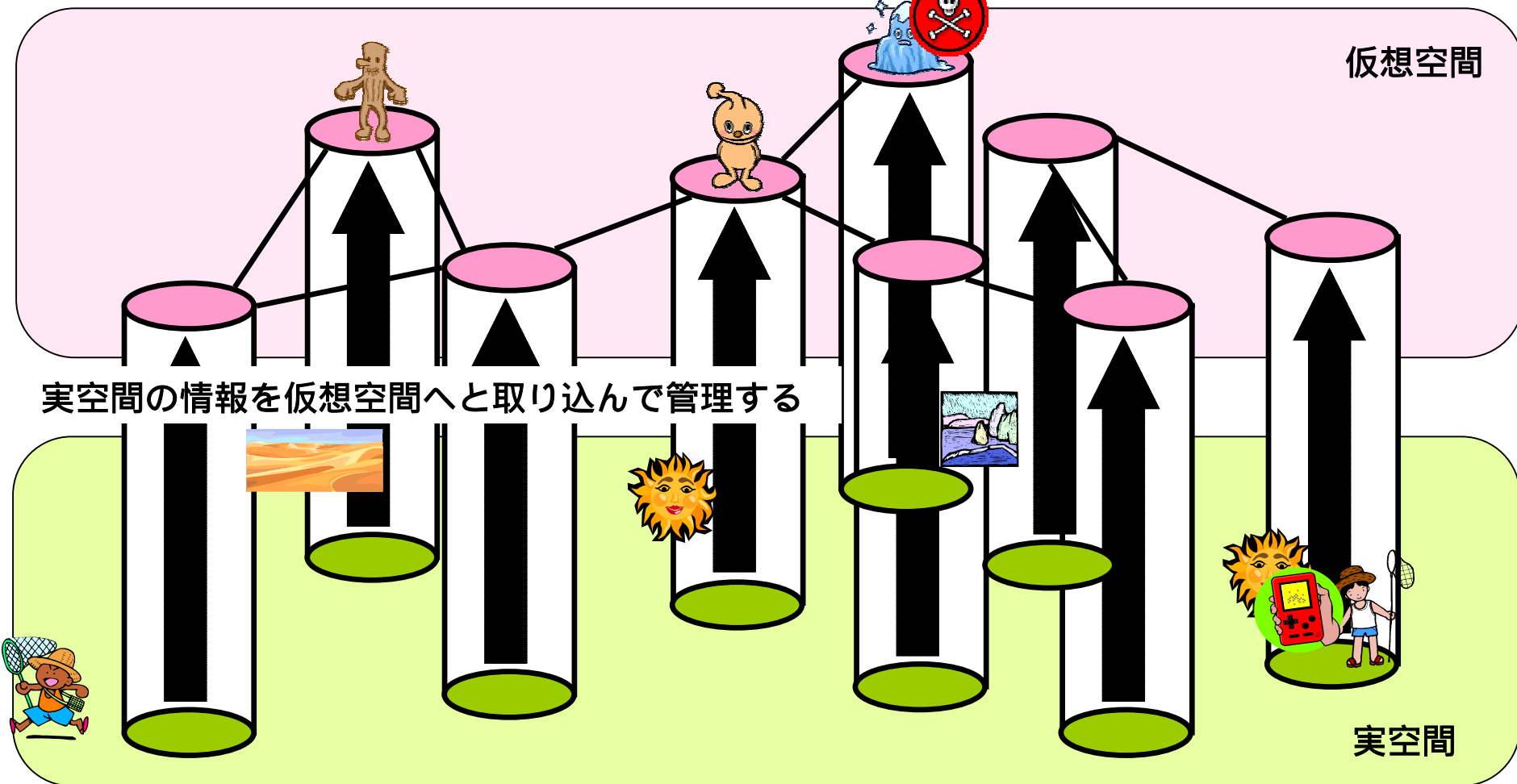
モンスターは仮想空間中で生息する

仮想空間

実空間の情報を仮想空間へと取り込んで管理する

実空間

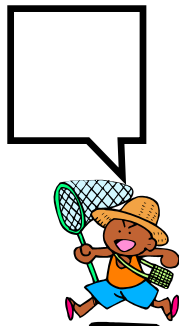
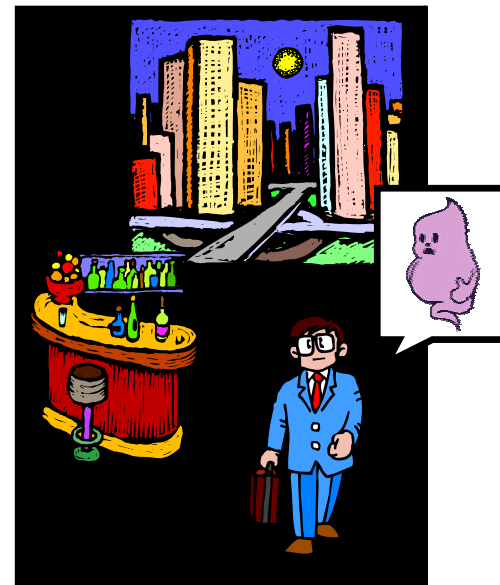
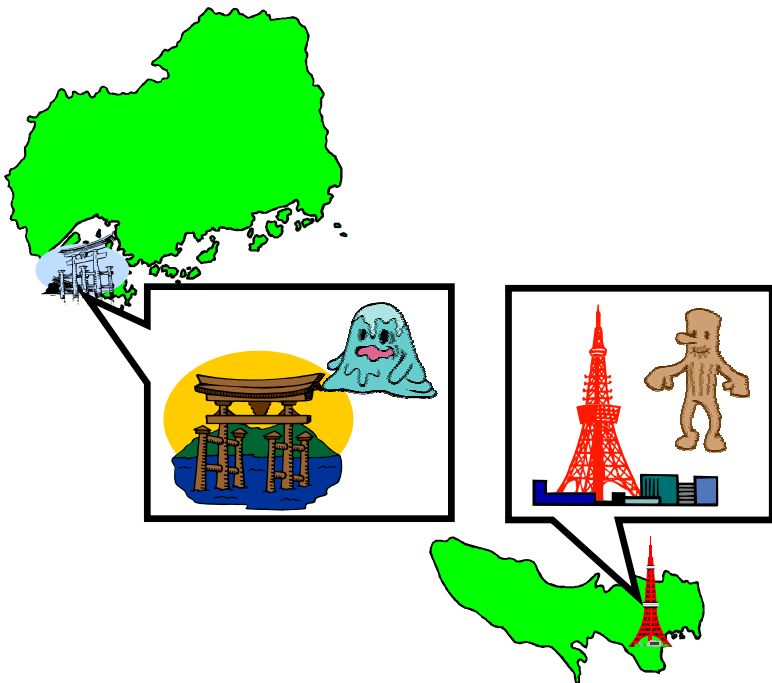
ユーザは実空間を移動してモンスターを捕まえる





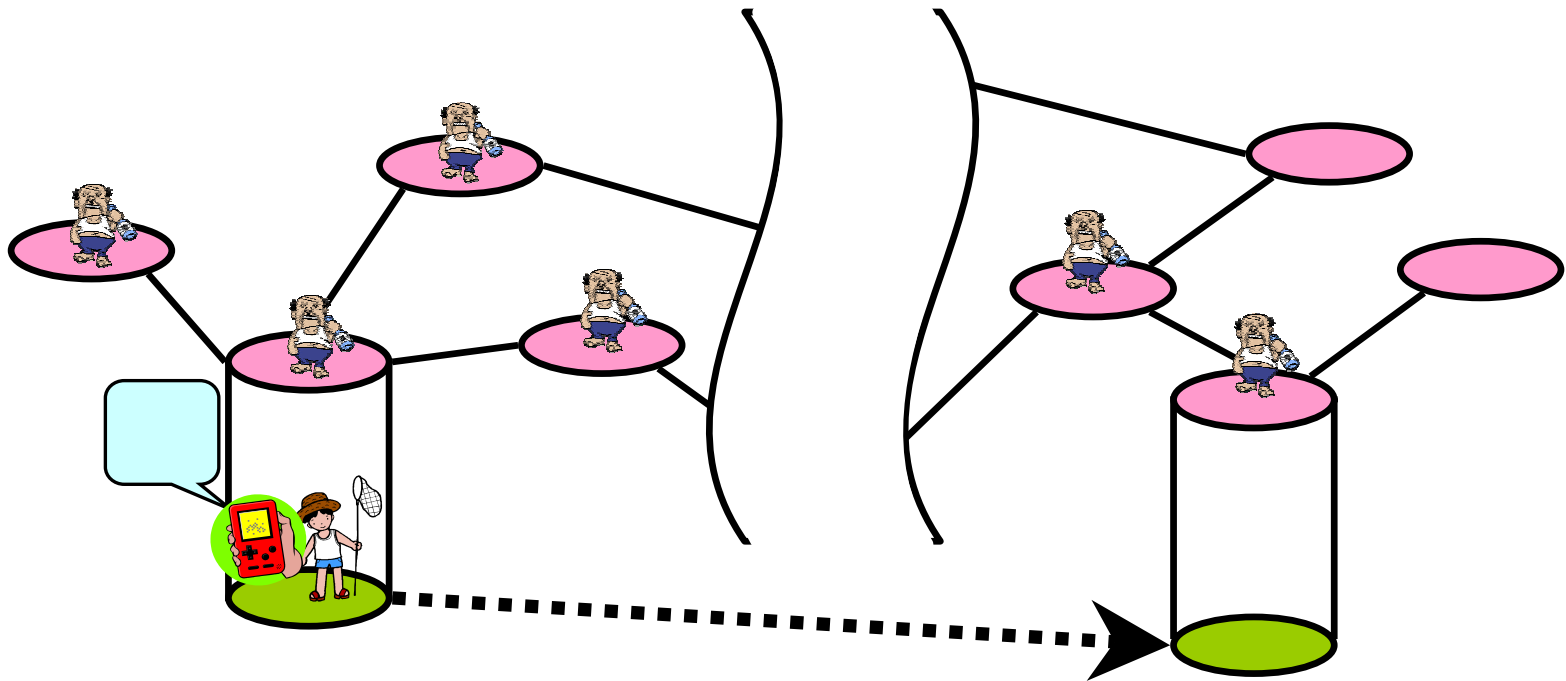
# 実践シナリオ例(1)

- 時間と場所を限定したモンスター
  - ご当地モンスター
    - イベント会場, 交通・旅行会社とのタイアップ
  - 夜中の歌舞伎町にしか登場しないモンスター
    - 子供たちは捕まえることが困難
    - 父親が捕まえて, 家で子供にプレゼント



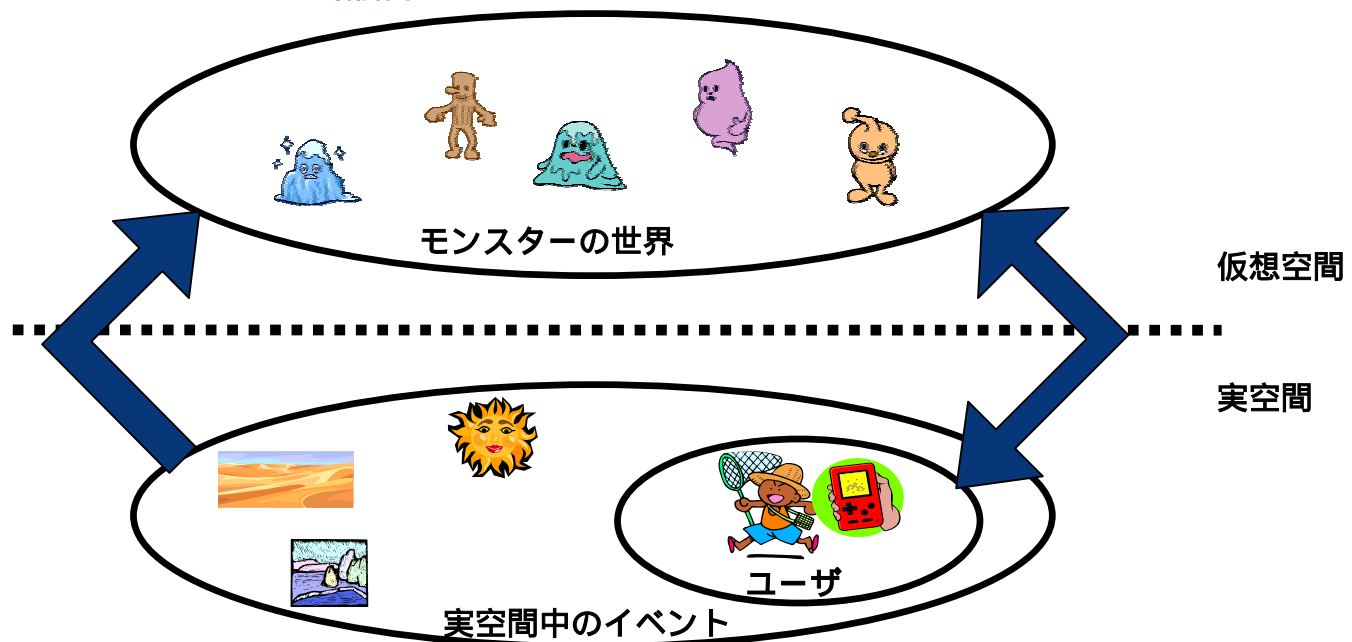
## 実践シナリオ例(2)

- ユーザによるモンスター生態系の変化
  - 特定の地域に生息するモンスター
    - 他の地域に移動しにくい
  - 捕獲したユーザが実空間中を移動
  - 遠隔地で仮想空間に戻す(リリース)
    - 遠隔地でモンスターが生息



# ユビキタスマンスターの実装

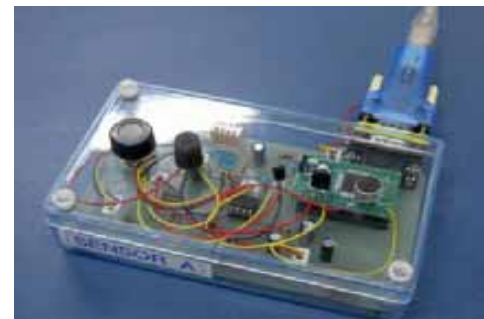
- ユビキタスマンスターの構成要素
  - 仮想空間中で生息するモンスター
  - 実空間中を移動してモンスターを捕獲するユーザ
  - 時々刻々変化する実空間の環境情報
- 構成要素間のインタラクション
  - モンスターと実空間
    - 実空間センサ情報に基づいた自律分散動作
  - モンスターとユーザ
    - ユーザによるモンスターの捕獲





# モンスターの自律分散動作の実装

- モンスターと実空間情報をモバイルエージェントとして実装
  - モンスターと実空間情報のインタラクションが可能
    - モンスターは生存エネルギーと環境に対する嗜好を持つ
    - 実空間情報はセンサの種類と値からなる
  - モンスターは実空間情報に応じてエネルギーが変化
    - 好みの環境ならば増え、逆ならば減る
    - エネルギーが増えたと、自己増殖
  - 実空間情報は実行環境中を伝播
    - モンスターは好みの環境を求めて移動
- センサアレイの実装
  - 実空間情報の取得
  - 6種類のセンサを搭載
    - 温度・湿度・明るさ・モーション・音・アルコール



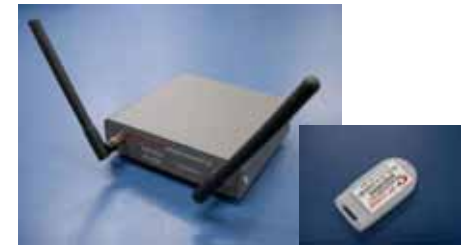
環境情報取得機構

# ユーザによるモンスターの捕獲の実装

- 網を振ってモンスターを捕獲する
  - 直感的な動作
    - 昆虫採集のメタファ
- モンスターネット
  - 加速度センサ搭載
    - micaを使用
- 捕獲の流れ
  - ユーザの接近をRFIDタグで検知
  - モンスターがゲーム端末に移動
  - モンスターネットを振る
    - モンスターごとに設定された捕獲条件の判定
      - 条件をクリアできるとモンスター捕獲成功
      - 条件がクリアできないとモンスターは戻ってゆく



モンスターネット



RFID

## まとめ

- ゲームはネットワークを扱うことでより豊かになる
- しかし、ネットワークはいつも土管として振る舞ってくれない
- ゲームの更新データの共有は低遅延でないといけない
- スケーラビリティが無いといけない
- このためにはユーザの利用ログ解析も有効
  
- これからは実空間情報を活用したゲームに期待

# 管理が手に負えない！

サービス提供者視点からの注目すべき変化

## 垂直統合型事業モデルから水平分業モデルへ

- 従来 1 社ですべてのサービスを提供していたが、昨今ではキャリア、ISP、ASPが同一物理網上に重畳してサービス提供

## 管理ドメインの量的増加

- キャリアが扱うべき網がコアネットワーク→アクセスネットワーク→カスタマーネットワークへと拡大し管理対象が増加

## 管理ドメインの質的拡大

- ホームネットワークやセンサネットの拡大によりサービスプロバイダが管理対象とする機器の種類も爆発的に増加
- 多様なトランスポートプロトコルを収容する方式

## 管理者の拡大

- 通信会社だけでなく、一般ユーザも自らの責任の範囲内で制御可能な親和性の高いネットワーク管理手法の必要性の増加



従来のSNMPベースの管理手法の破綻

# 「新世代」の ネットワーク管理手法の研究

## 目的

- ネットワークが垂直・水平的に様々な事業者が存在することを前提に、物理層からアプリケーションまで柔軟に管理可能なネットワーク管理アーキテクチャの確立

## 目標

- 従来のIP網に加え、センサネットワークやホームネットワークを管理可能なネットワーク管理プロトコルとコントロールプレーンの設計
- SNMP, CLIの限界を乗り越え、ユーザの手で分散的に配置されたエンティティを分散的に管理・制御可能なエンティティ管理機構とデータモデルの設計

# アプローチ

- ネットワーク管理をする権利はネットワーク管理者の特権ではない！
  - まずはプログラマにも親和性が高くあるべき
    - その次ユーザ
  - Webマッシュアップ技術にヒントを得る
- RESTアーキテクチャスタイルに基づく管理機能 (Active Proxy) を導入
  - Active Proxyでプロトコル差異を吸収し新しく登場するネットワークにも対応
  - 分散調整を持たせSNMPのMIBにみられる中央集権的かつ固定的な管理を排除
  - APIで機器を制御可能に

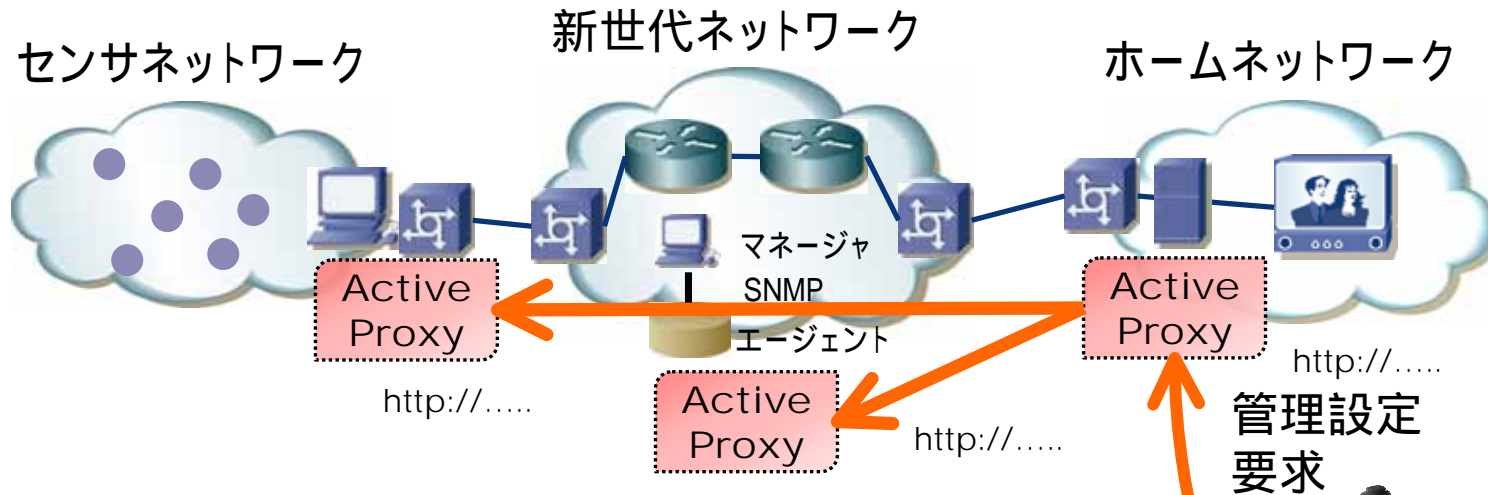


ネット  
ワーク  
管理者

プログラ  
マ

ユーザ

# REST APIによるネットワーク管理



Active Proxy

RESTアーキテクチャスタイルに基づく異種ネットワーク機器管理機能  
各ネットワークに一つ以上おかれ、専用プロトコルとHTTPのProxyとして働く

ユーザがXMLによって管理設定要求を設定するとActive Proxy間で連携し遍在するリソースが連携して設定管理される。



ユーザ