

コミュニティにおけるアシュアランス性 実現のための自律サービス発見技術

堀越 悠久史, Khaled Ragab, 大野 敬典, 梶 直弘, 森 欣司

東京工業大学大学院情報理工学研究科計算工学専攻

〒152-8550 東京都目黒区大岡山 2-12-1

TEL: 03-5734-2664 FAX:03-5734-2510

E-mail: {horikoshi@mori., ragab@mori., nouri@mori., nkaji@mori., mori@}cs.titech.ac.jp

あらまし 近年、ますますユーザの嗜好、利用場所や時間に応じた情報サービスが求められている。我々は、嗜好や状況を共通に持ったユーザ群であるローカルマジョリティを考慮し、ローカルマジョリティの要求に応えるサービスを提供するための自律コミュニティシステムを提案している。自律コミュニティシステムにおいては、サービスはネットワーク上に配置され、その提供位置がユーザの要求に合わせて時々刻々と変化する。本稿では、ユーザからサービスを要求するときのサービス発見について、サービス位置が予測不可能な状況下でも応答性を悪化させることなくサービス発見できることを保証する適応的なサービス発見技術を提案し、評価を行なった。

キーワード ローカルマジョリティ, コミュニティ, サービス発見

Autonomous Service Discovery Technology To achieve Assurance in Community

Yuji Horikoshi, Khaled Ragab, Takanori Ono, Naohiro Kaji, and Kinji Mori

Tokyo Institute of Technology

2-12-1 Ookayama, Meguro, Tokyo 152-8552, Japan

TEL: +81-3-5734-2664 FAX: +81-3-5734-2510

E-mail: {horikoshi@mori., ragab@mori., nouri@mori., nkaji@mori., mori@}cs.titech.ac.jp

Abstract In recent years, information services in accordance with users' preference, utilization place and time are required. Considering the *local majority*, which consists of users who have shared preferences or are in similar situations, autonomous community system is proposed to provide services according to the requirements of the local majority. In this autonomous community system, services are allocated on the network and changing their positions from time to time coping with the change of users' requirements. This paper proposes a service discovery technology that guarantees to find the services with the assured no-worse response time even in condition of unpredictable service positions, and proves the effectiveness of the proposal with an evaluation.

Key words Local Majority, Community, Service Discovery

1. はじめに

近年、移動体通信技術の発達により、グローバルで多種多様なサービスを「いつでも・どこでも・誰でも」利用できる環境になってきているが、一方で、提供されるサービスが膨大な数に及び、ユーザはどのサービスが自分の要求を満たすものかが分からなくなっている。同時に、情報サービスを利用するユーザが増加し、多様なニーズをユーザが持つことで、サービス提供側もユーザに対して適切なサービスを提供することが困難になってきている[2]。このように、「今だから、ここだから、あなただけ」と言うような、ユーザの場所や利用時間、ユーザのニーズを考慮した情報サービスシステムが求められる。

我々は、特定の嗜好や状況を共通して持つユーザ集団(ローカルマジョリティ)に対して、ローカルマジョリティ内におけるユーザ間のインタラクションや、サービスプロバイダ(SP)とローカルマジョリティとの間のインタラクションによって柔軟なサービスを利用・提供することが可能となる自律コミュニティシステムを提案している。

自律コミュニティシステムにおいて、ユーザからサービスを要求するとき、自律コミュニティシステム上で提供されるサービスを探索・発見することになる。このサービス発見のための技術として、提供されるサービスの位置が予測不可能であってもサービスの発見を保証し、その応答性を悪化させない自律サービス発見技術を提案する。また、シミュレーションによって本提案技術の適切な動作の検証および応答性について検証を行なった。

2. ニーズ

2.1 アプリケーションニーズ

ユーザのその場の状況、利用時間や利用場所に適合したサービスを提供するものとして、ユーザの持つ携帯情報端末に対して行なうサービスが考えられる。このようなサービスでは、ユーザのニーズや状況に見合った情報提供を行なうため、ローカルマジョリティに応じて情報提供エリアを柔軟に変えることが必要となる。

また、円滑に素早くサービスを提供する事が求められる。

2.2 システムニーズ

ユーザの利用時間や利用場所といった状況に合わ

せてサービス提供エリアを柔軟に変える必要があるため、システムには柔軟性が求められる。同時に、いち早いサービスの提供と利用が求められる事から、システムには応答性が求められる。

3. 自律コミュニティシステム

2.2で述べたシステムニーズを満たすものとして、自律コミュニティシステムを提案する。自律コミュニティシステムは柔軟性を満たすため、自律分散システム[1]を基礎とするシステムである。

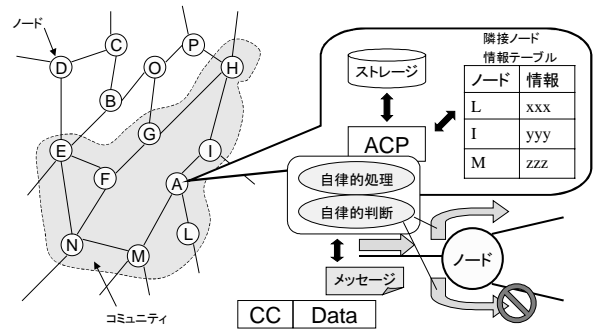


図 1: 自律コミュニティシステムアーキテクチャ

3.1 アーキテクチャ

自律コミュニティシステムのアーキテクチャを図1に示す。アーキテクチャは、データフィールドアーキテクチャを用い、内容コード通信によって各コミュニティノードが自律的に通信、処理を行なう。各ノードは局所的な情報に基づいて判断・処理を行なう ACP(自律管理機能)、メンバの要求情報やサービスを蓄えるストレージ、隣接ノード情報テーブルから構成される。

自律コミュニティシステムで送受信されるメッセージには受信者を指定するアドレス情報などは含まれておらず、受信する側は内容コード(CC)を見てメッセージを選択受信する。図1で示したとおり、各ノードは隣接ノードに対してリンクを持っており、隣接ノード情報テーブルでそれを管理している。各ノードは、自ノードに送信されてきたメッセージを自律的判断によって隣接ノードに転送するかどうかを決定する。その結果、メッセージが転送されたエリアがコミュニティであり、それがコミュニティ内データフィールドとなる。コミュニティ内データフィールドに対してブロードキャストを行なうにはフラディングの手法を用いる。このようなコミュニティ内データフィールドを利用することで、システムに求められる柔軟性を実現する。

3.2 アプリケーション例

自律コミュニティシステムのアプリケーションシステムとして、時間距離指向情報サービス[3][4][5]が提案されている。時間距離指向情報サービスでは、ユーザの SP への物理的アクセス時間をユーザから SP までの時間距離と定義し、時間距離が一定範囲内のエリアのユーザにのみ SP の情報を配布し、オンラインでのマーケティングを行なう。この際、ノードは携帯情報端末のための無線基地局に対応され、基地局間の時間距離と SP から配布される情報の有効時間によってコミュニティのエリアが決定される。

4. 自律サービス発見技術

4.1 サービスの発見

自律コミュニティシステムにおいて、ユーザが現状で提供されるサービスでその要求が満たされない場合、属性情報などをもとに自身の要求に合ったサービスを探索し、発見する必要がある。例えば、その場に居合わせた人同士を結びつけてコミュニティを形成するマッチングサービスなどが考えられる。

3.1で示したとおり、自律コミュニティシステムのアーキテクチャは多数のコミュニティノードがリンクによって接続され、ネットワーク化されている。ここで提供されるサービスは膨大な数に及び、それぞれの提供位置も変化する。このように、提供エリアが変化したり、サービスが移動したりすることにより、サービスの位置が予測不可能な状況となる。

このようなサービスの位置が予測不可能な状況においても、ユーザの要求に見合ったサービスを提供することを保証する技術が求められる。

4.2 従来のサービス発見技術と問題点

状況や嗜好に応じたサービスの発見・提供するには、サービス要求者のノードからフラッディングによりサービス発見要求を送出し、探索する。また同時に、ネットワーク全体に対して影響を与えないようにするため、要求メッセージに TTL(Time To Live)を設定して、要求ノードの周囲一定範囲を探索するようにする。その一定範囲内で見つからなければ、TTL を大きく設定して再び要求送出行ない、見つかるまでこれを繰り返すことでサービスの発見が行なわれる。

しかし、大規模なネットワークで、かつサービス提供位置が予測不可能であるため、適切な TTL を決定するのは困難である。また、要求を送出した後、

一定時間の間、サービス提供ノード(リプライヤノード)からのリプライを待ち、リプライがなければ TTL を大きくして再度要求を送出するわけであるが、この再要求待ち時間を決定することも困難である。そのような状況下において、この手法でサービス発見を行なうと、発見に失敗したことによる再要求待ち時間により、サービス発見の応答性が悪化すると考えられる。

このような予測不可能な状況でも、応答性を悪化させずにサービスの発見を保証する技術が求められる。

4.3 自律サービス発見技術の概要

予測不可能な状況下でサービス発見を行なう際の応答性悪化を防ぐため、自律サービス発見技術を提案する。なお、仮定として、リプライヤノードが必ず存在するものとし、リンクやノードのフォルトは起こらず、またリンクはすべて双方向リンクとする。

4.3.1 キーアイデア

提案技術は、事前にサービス発見要求の到達範囲を決定するのではなく、サービスが発見できた後でそのサービス発見要求を取り消すようにし、ネットワーク全体への影響を与えなくするものである。このようにして、一度のサービス発見要求を送出するのみでサービスを発見できることを保証でき、再要求待ち時間によるサービス発見の応答時間の悪化を防ぐことができると考えられる。

提案技術のコンセプトは、要求メッセージの伝搬を、ネットワークを伝わっていく情報の波と捉えるものである。このとき、進行速度の違う2つの波を1組と考え、一つをサービス発見要求メッセージ、もう一つをサービス発見リプライメッセージとする。進行速度の遅い波をサービス発見要求の波とし、進行速度の速い波をサービス発見リプライの波とする。また、これらの波は、出会うと双方ともに消滅するものとする。

今、要求ノードからサービス発見要求の波が送出され、それがリプライヤノードに到達すると、リプライヤノードからサービス発見リプライの波が送出される。要求の波とリプライの波が両方とも広がっていくが、リプライの波の方が速いため、いずれ要求の波に追いつき、ある時点で双方とも消滅する。つまり、リプライの波は要求の波をキャンセルする機能を持っている。

情報の波とは、ノード間で伝達されるメッセージ

がネットワーク内を広がりながら伝搬していくものである。波の速度を実現するものとして、ノードが要求メッセージを転送する前に待ち時間(要求転送待ち時間)を設けるものとする。すなわち、その待ち時間が大きければ波の速度は遅く、待ち時間が小さければ波の速度は速いものとなる。

4.3.2 サービス発見の過程

自律サービス発見技術の全体の流れを図 2 に示す。まず、要求ノードから要求が送信され(a)、それがネットワーク内を伝搬してリプライヤノードに到達し(b)、リプライが発せられる(c)。要求とリプライ双方がネットワーク内を伝搬していく(d)が、リプライの方が速いため、いずれ要求に追いつき、要求が取り消されてそれ以上のメッセージの伝搬が起こらなくなる(e)。

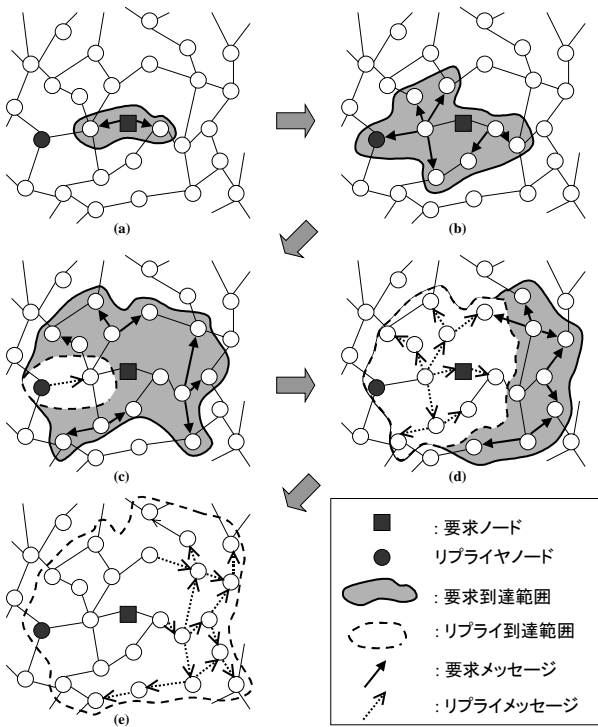


図 2: サービス発見の過程

要求の波の速度よりリプライの波の速度の方が速ければ、必ず要求の波は取り消されるが、リプライの波がネットワーク全体に伝播する恐れがある。これを防ぐため、各ノードが未受信の要求に対するリプライを受信した場合にはそのリプライを破棄する。

要求の波の速度が小さければ、当然サービス発見までの応答性は悪化するが、要求の波の速度が大きいと、追いつかれて消滅するまでの時間がかかり、結果的にトラフィック量の増大を招く。波を伝達するノードが要求ノードとの距離などをもとにして自

律的にその待ち時間を決定することで、これに対処する。

4.4 メッセージフォーマット

自律サービス発見技術で用いられる 2 つのメッセージを示す。

4.4.1 要求メッセージ

サービス発見要求メッセージのメッセージフォーマットを図 3 に示す。

- 内容コード: Content Code
このメッセージがサービス発見要求メッセージであることを表す。
- 要求ノード ID: Requester Node ID
このメッセージを送信したノードの識別子。
- メッセージ ID: Message ID
要求ノードで一意に付与されるメッセージ識別子。要求ノード ID とこのフィールドでメッセージを一意に識別する。
- ホップ数: Number Of Hops
メッセージのホップ数を表す。ノードが要求メッセージを次ノードに転送する際に、その待ち時間を決定する指標となる。
- 要求内容: Request Content
どのようなサービスを発見・利用したいのかを表す。このフィールドでユーザがどのようなサービスを要求しているのかを記述する。また、ノードはこのフィールドを見て、自身がサービスを要求に対してリプライ可能かどうかを判断する。

Content Code (CC)	Requester Node ID	Message ID	Number Of Hops	Request Content
-------------------	-------------------	------------	----------------	-----------------

図 3: 要求メッセージのフォーマット

4.4.2 リプライメッセージ

サービス発見リプライメッセージのメッセージフォーマットを図 4 に示す。

- 内容コード: Content Code
このメッセージがサービス発見リプライメッセージであることを表す。
- 要求ノード ID: Requester Node ID
このメッセージに対応するサービス発見要求メッセージがどのノードが作成したものなのかを示す。
- リプライ対象メッセージ ID: Request Message ID
このメッセージが、どのサービス発見要求メッ

メッセージに対するリプライなのかを示す。要求ノードIDとリプライ対象メッセージIDでメッセージが一意的に識別する。

● リプライヤノード ID: Replyor Node ID

メッセージをリプライしたノードの識別子。ここを見ることで、リプライメッセージを受けた各ノードは、サービスが提供されるノードがどれであるかを知ることができる。

● リプライ内容: Reply Content

リプライの内容。提供されるサービスによって内容は異なる。

Content Code (CC)	Requester Node ID	Request Message Message ID	Replyor Node ID	Reply Content
-------------------	-------------------	----------------------------	-----------------	---------------

図 4: リプライメッセージのフォーマット

4.5 ノードの基本動作

各ノードは、一つ一つのサービス発見要求に対し、その状況に応じて5つの状態を取る。このため、各ノードはサービス発見要求とその状態とを対応づける状態テーブルを持っている(図5)。また、メッセージをどの隣接ノードに転送すべきかを記憶する機能も状態テーブルに含まれる。

各ノードはメッセージを受信すると、そのメッセージの内容コードを見て、そのメッセージがサービス発見要求メッセージかサービス発見リプライメッセージであった場合、各ノードはそのメッセージに対応する状態に応じて動作を行なう。状態によってはタイマを持つものもあり、タイムアウトしたときにもその状態に応じた動作を行なう。

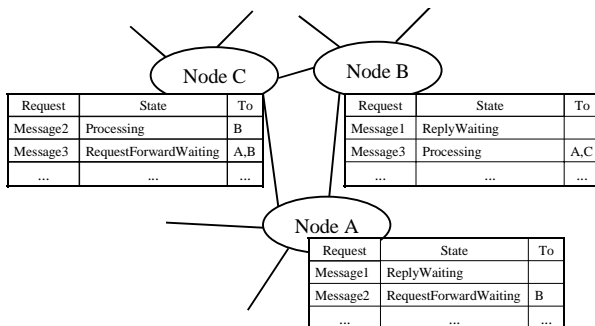


図 5: ノードの状態テーブル

4.6 ノードの状態遷移

図6に、自律サービス発見技術を実現するための各ノードの状態遷移図を示す。図6中のRequestはサービス発見要求メッセージ、Replyはサービス発

見リプライメッセージである。また、図7に図2で示したサービス発見の過程の中での状況と図6で示した各ノードの状態との対応を示す。

図7にあるように、状態遷移はサイクルのようになっている。すなわち、サービス発見要求メッセージを受信していない通常状態から始まって、要求の受信により要求処理を行ない、応えられなければ次ノードに転送するが、その前に一定時間待ってから次ノードに要求の転送を行なう。その後、リプライが来るまで何もせず、リプライを受信したらリプライを次ノードに転送し、メッセージ吸収状態に遷移する。その後一定時間を経て通常状態に戻る。関係したすべてのノードが通常状態へ戻ることで、提案技術のアルゴリズムは終了する。

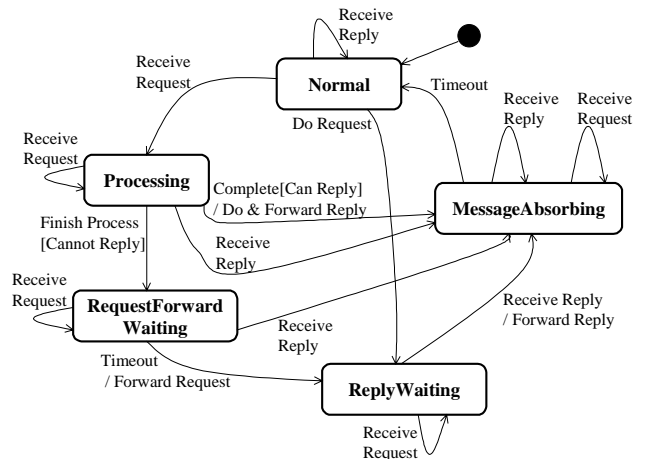


図 6: ノードの状態遷移図

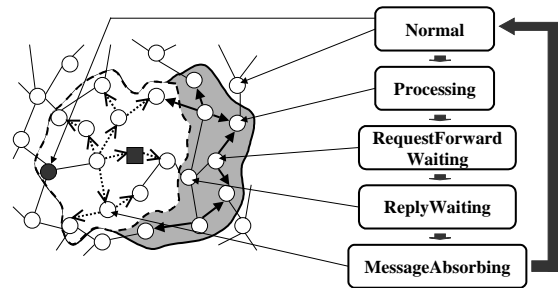


図 7: ノードの状態とサービス発見の過程

以下、各状態について詳しく述べる。

4.6.1 通常状態: Normal

サービス発見要求メッセージをまだ受信しておらず、状態テーブルにサービス発見要求メッセージが登録されていない状態である。無関係なサービス発見リプライメッセージを受信しても何も行なわないため、結果としてサービス発見要求メッセージが到達した範囲にのみサービス発見リプライメッセージ

が伝搬するようになり、サービス発見リプライメッセージが永遠に伝搬するのを防ぐ。

- サービス発見要求メッセージ受信
 1. 状態テーブルにメッセージを登録、転送先として送信元を除くすべての隣接ノードを指定
 2. 要求処理状態に遷移
- サービス発見リプライメッセージ受信
 1. 受信メッセージを破棄
- 新たにサービス発見要求を送出
 1. サービス発見要求メッセージを作成し、すべての隣接ノードに送信
 2. リプライ待ち状態に遷移

4.6.2 要求処理状態: Processing

自ノードがサービス発見要求メッセージに応じてリプライすることができるかどうかを処理・判断するための状態である。

- 処理の結果、リプライ可能と判断
 1. サービス発見リプライメッセージを作成
 2. 隣接ノードすべてにサービス発見リプライメッセージを送信
 3. メッセージ吸収状態に遷移
- 処理の結果、リプライ不可能と判断
 1. 要求転送待ち状態に遷移
- 処理途中でサービス発見要求メッセージ受信
 1. 状態テーブルの該当メッセージの転送先リストから今回の送信元ノードを削除
 2. 処理を続行
- 処理途中で処理中の要求に対応するサービス発見リプライメッセージを受信
 1. 処理を中断
 2. サービス発見リプライメッセージを送信元ノード以外のすべての隣接ノードに転送
 3. メッセージ吸収状態に遷移

4.6.3 要求転送待ち状態: RequestForwardWaiting

サービス発見リプライメッセージが受信されないかどうかを一定時間待ち、その間に受信されない場合は次ノードにサービス発見要求メッセージを転送する状態である。受信された場合にはリプライが追いついたという事であるため、両メッセージを破棄する。

この状態では、要求転送待ち時間 T_{Wait} が存在し、 T_{Wait} の間だけノードはサービス発見リプライメッセージが受信されないかどうかを待つ。 T_{Wait} はノードによって自律的に決定される。例えば、要求ノ

ードから自ノードまでのホップ数などによってその大きさを決定する。

- T_{Wait} 間にサービス発見要求メッセージ受信
 1. 状態テーブルの該当メッセージの転送先から今回の送信元ノードを取り除く
 2. 待機処理を続行
- T_{Wait} 間にサービス発見リプライメッセージ受信
 1. ノードが保持する両メッセージを破棄
 2. メッセージ吸収状態に遷移
- T_{Wait} がタイムアウト
 1. 状態テーブルに記録された転送先ノードに対してサービス発見要求メッセージを転送
 2. リプライ待ち状態に遷移

4.6.4 リプライ待ち状態: ReplyWaiting

サービス発見リプライメッセージが受信されるのを待ち、受信された際にはそのサービス発見リプライメッセージを次ノードに転送する状態である。

- 同一のサービス発見要求メッセージを受信
 1. メッセージを破棄
- 対応するサービス発見リプライメッセージを受信
 1. メッセージの送信元ノードを除いたすべての隣接ノードにリプライメッセージを転送
 2. メッセージ吸収状態に遷移

4.6.5 メッセージ吸収状態: MessageAbsorbing

一足飛びに通常状態に戻るのを避け、緩衝材としてはたらく状態である。一足飛びに通常状態に戻った場合は、通常状態に戻った後に違うルートを通った同一のサービス発見要求メッセージを受信し、再び状態遷移のサイクルを回る事になって全く無駄な通信を生じさせることになる。

この状態では、メッセージ吸収時間 T_{Absorb} が存在し、 T_{Absorb} の間だけノードは受信したメッセージをすべて破棄する。

- T_{Absorb} の間にメッセージを受信
 1. メッセージを破棄
- T_{Absorb} がタイムアウト
 1. 状態テーブルからメッセージを削除
 2. 通常状態に遷移

5. シミュレーション・評価

シミュレーションの目的は、提案技術の動作検証および、応答性の評価である。また、提案技術のトラフィック量と応答性に関する特性を明らかにする。

本シミュレーションの前提となるモデルを示す。ネットワークのトポロジとして六角形のメッシュネットワークを用いる。また、サービス発見・要求取り消しの様子をスクリーンに表示させる(図 8)。図中の番号は対応するノード番号を示す。図 8 では、ネットワーク半径 4 で 37 ノードしか描かれていないが、実際には図 8 と同様のノード配置のされ方で、より多数のノードが配置される。シミュレーションでは、ネットワーク半径 35 で 3781 個のノードが含まれるネットワークを使用した。

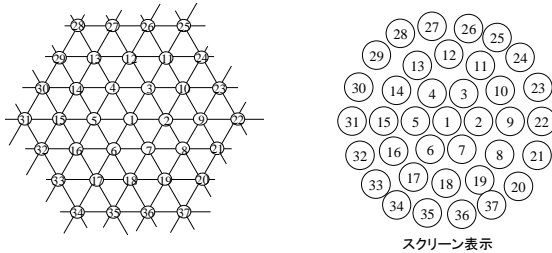


図 8: ネットワークトポロジ

すべてのシミュレーションに共通するパラメータを示す。リンクの伝送遅延時間 T_{Delay} を 1(ms) とする。ノードの要求処理状態の処理時間 $T_{Process}$ を 5(ms) とする。ノードのメッセージ吸収状態の吸収時間 T_{Absorb} を 50(ms) とする。

5.1 動作検証

図 9 に、要求転送待ち時間 T_{Wait} を 8(ms) とし、リプライヤノードとの距離を 5 とした時の提案技術の動作の様子を示す。ネットワークの中心の黒いノードが要求ノードで、その左下にある黒いノードがリプライヤノードである。図 9 より、提案技術が適切に動作したと分かる。

5.2 応答性の評価

リプライヤノードとの距離を変化させたときのサービス発見の応答時間について述べる。各ノードのリクエスト転送待ち時間 T_{Wait} を 2,4,8(ms) とした。また、TTL を定めて発見できなかった場合に再度要求を行なう方式との比較も行なった。TTL の増やし方を 5,10,15 としたものととの比較を示す。その再要求待ち時間は、要求到達範囲内のすべてのノードが要求処理を終えるまでの時間を計算し、その時間とリプライが要求ノードに到達するまでの時間とを加算したものである。図 10 にこれを示す。

TTL を定めて発見できなかった場合に再要求を行なう方式では、次ノードに転送する際に待ち時間が存在しないため、リプライヤノードとの距離が近い

範囲では応答が良好である。しかし、再要求が行なわれる場合には再要求待ち時間によって大幅に応答時間が悪化することが分かる。提案技術ではリプライヤノードとの距離と応答時間が線形の関係にあることが分かる。これは要求メッセージの再送を行なうことが無いからであり、従って、提案手法はサービス位置が予測不可能な状況に対して、応答性について適応性を持つことが分かる。

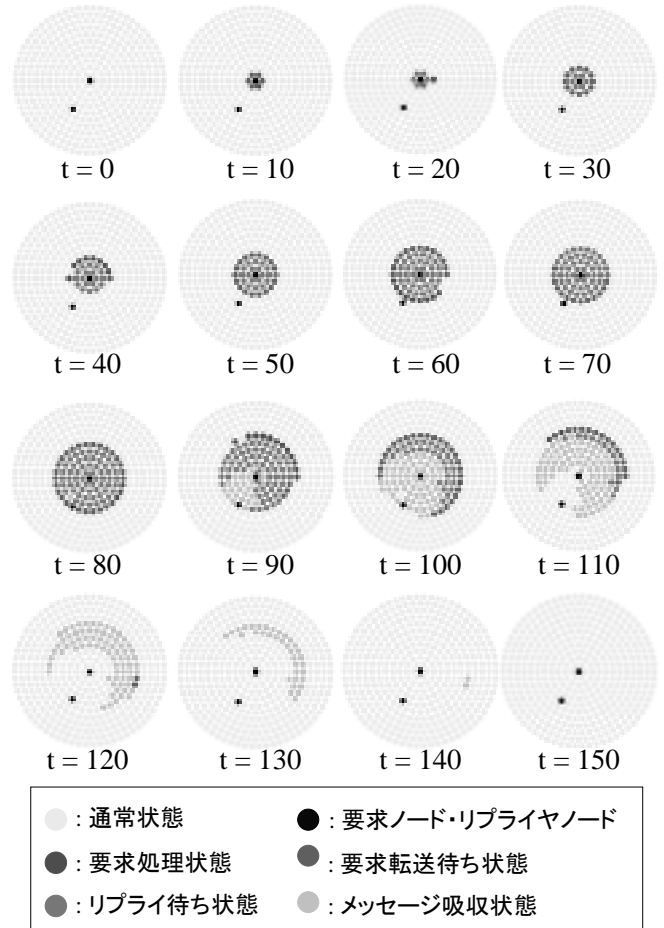


図 9: 提案技術の動作結果

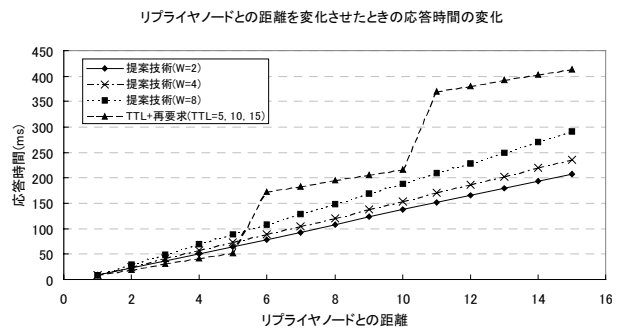


図 10: リプライヤとの距離による応答時間

5.3 要求転送待ち時間による特性

要求転送待ち時間を変化させたときの提案技術の

応答時間および総メッセージ数について述べる。リプライノードとの距離(ホップ数)は 5 ホップ、10 ホップとした。図 11 にこれを示す。

図 11 から、提案技術は応答時間と総メッセージ数はトレードオフの関係にあることが分かる。要求転送待ち時間が大きければ、それにより要求がリプライノードに届くことが遅くなり、応答時間が悪化する。要求転送待ち時間を大きくすると逆に総メッセージ数が小さくなる。これは、要求が広がっていくときに、広がりが大きくなる前にリプライが要求に追いついて要求の伝達がそれ以上起こらないことによるものである。しかし、総メッセージ数は一定以上には小さくならない。これは、要求がリプライノードに到達した後、その要求が次のノードに転送される前にリプライが追いついてきたということによるものである。

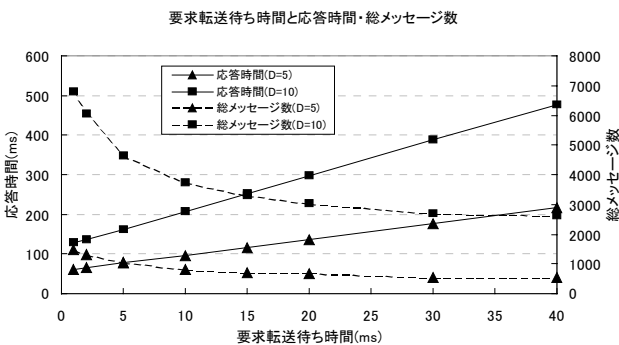


図 11: 要求転送待ち時間による特性

5.4 ノードごとに要求転送待ち時間を自律的に変化させるときの特性

各ノードの要求転送待ち時間 T_{Wait} をノードごとに自律的に変化させた場合の特性を示す。 T_{Wait} の求め方は以下の通りである。

$$T_{Wait} = \begin{cases} Hops & Hops < 8 \text{ のとき} \\ 7 & Hops \geq 8 \text{ のとき} \end{cases} \quad Hops: \text{ホップ数}$$

図 12、13 に、以上で述べた条件での応答時間と総メッセージ数を示す。また、要求転送待ち時間を 4 と 8 で固定させた場合の特性も示す。図 12、13 より、要求転送待ち時間を各ノードで変化させた場合の応答時間は、要求転送待ち時間を 4 で固定した場合の応答時間と近い値を示し、また総メッセージ数は、要求転送待ち時間を 8 で固定した場合の総メッセージ数と近い値を示している。従って、それぞれのノードの自律的判断により転送待ち時間を動的に適切に変化させた場合、応答性が向上し、またネットワークの帯域を有効に利用することが出来ると

言うことが示された。

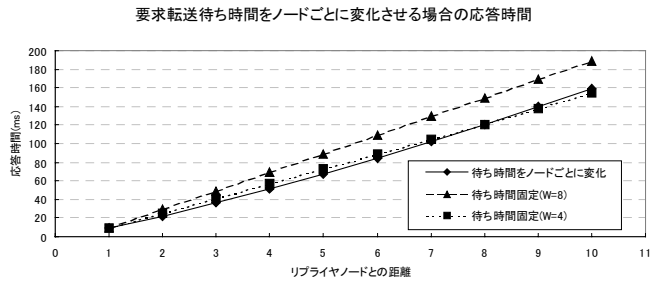


図 12: 要求転送待ち時間をノードごとに变化させるときの応答時間

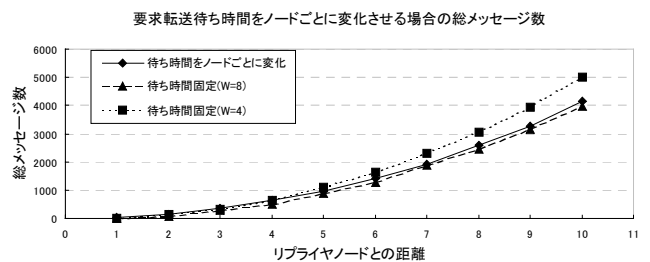


図 13: 要求転送待ち時間をノードごとに变化させるときの総メッセージ数

6. おわりに

本稿では、利用場所や利用時間に応じた情報サービスシステムのための自律コミュニティシステムにおいて、サービス発見のための技術として自律サービス発見技術を提案し、シミュレーションによって、サービス位置が予測不可能な状況下においても応答性が悪化しない適応的な技術であることが示された。

参考文献

- [1] Kinji Mori, "Autonomous Decentralized Systems: Concept, Data Field Architecture and Future Trends", Proc of ISADS93, pp150-157, March 1999
- [2] Kinji Mori et al., "Service Accelerator(SEA) System for Supplying Demand Oriented Information Services", Proc. of IEEE CS Workshop Future Trend of Distributed Computer Systems, pp149-161, 1994
- [3] 大野 敬典, 森 欣司, "アシュアランス性実現のための利用場所と時間を考慮した情報配布技術," アシュアランスシステムシンポジウム 論文集, pp.114-119, June.2001.
- [4] ソーサワン センサティット, 鈴木 泰山, 森 欣司, "特定エリア不特定多数の要求を反映したサービス情報選択技術," 電子情報通信学会 2001年総合大会, pp 143, March 2001.
- [5] Naohiro Kaji, Khaled Ragab, Takanori Ono and Kinji Mori, "Autonomous Synchronization Technology for Achieving Real Time Property in Service Oriented Community System," Proc. of The 2nd International Workshop on Autonomous Decentralized System (IWADS 2002), pp16-21, November. 2002