

無線アドホックネットワーク下で協調的な意思決定を 実現する Noroshi アーキテクチャの提案

A Proposal of Noroshi Architecture for Collaborative Decision making
under Ad hoc Wireless Network

渡 邊 充 隆

要 約 近年、無線通信が可能な電子機器が増えてきており、その数や種類は今後も増加の一途を辿ることが予想される。無線アドホックネットワークを利用すれば、いつでもどんな場所でも即時にネットワークを構築し、P2Pアーキテクチャで相互に連携するようなシステムを実現することができる。しかし、膨大な数の機器を連携させようとした場合、それらを矛盾無く連携させるためには、機器間で協調的な意思決定を行えるような仕組みが必要となる。そこで本稿では、無線アドホックネットワーク環境下で協調的な連携を行うための仕組みである Noroshi アーキテクチャ^{*1}を提案する。

Abstract Various types of wireless communication enabled devices are on an upward trend. With wireless ad hoc networking and P2P technology, collaborative applications can be implemented on an anytime, anywhere basis by building on demand P2P networking. However, in order to interface an enormous number of devices with wireless network consistently, the mentioned applications should have a mechanism for collaborative decision making. In this paper, we propose "Noroshi Architecture" realizing collaborative decision making under wireless ad hoc networking.

1. はじめに

ユビキタスコンピューティングを実現するためのさまざまな要素技術が確立してきている。そのような要素技術の中で注目されているものに無線通信技術とP2P（ピア・ツー・ピア，端末間直接通信）がある。両者はもともと独立して発展してきたが、近年、それらを組み合わせて、屋外など、システムのためのインフラストラクチャが存在しない状況や、常に移動しているような状況においても利用可能なシステムを構築しようという試みが多く見られるようになってきた。

無線通信技術の発展にともなって低価格化してきた無線機器は、PCやPDAといった人間が直接使うものだけではなく、ユーザインタフェースを持たない超小型のセンサデバイスのようなものにまで、既に組み込まれている。実際、カリフォルニア大学バークレイ校などでは無線通信可能な組み込み機器に関する研究プロジェクト WEBS¹⁾の一環として、各種センサを搭載できるアドホック無線センサ端末兼中継モジュール Mica MOTE²⁾に関する研究が盛んに行われている。あらゆるものに無線機器が組み込まれれば、それぞれが他のデバイスと通信し、互いに連携し合いながら、一つのシステムを形作るということが可能になってくる。さらに、アドホックネットワーク（アクセスポイントなどを必要とせず、無線通信可能なデバイスだけで構成される）即席のネットワークを用いて互いに通信ができるため、屋外での催事や災害時の救援活動など、その場限りの局所的なシステムを構築するのに有効である。局所的なシステムでは、即時的な構築が重視されるので、サーバによる各端末の集中的な制御・管理は適していない。そこで、局所的なネットワークにおいても、サーバを必要とせず各ノードが互いに連

携できる仕組みが望まれる。特に、多くの機器が接続しあうシステムの場合は、機器間の連携に矛盾が生じないような配慮が必要である。

一方 P2P アーキテクチャは、ADSL 環境などによるインターネットへの常時接続環境の急速な普及に伴い、その接続端末が爆発的に増加し、伝統的なクライアント/サーバアーキテクチャに性能的な限界が見え始めたことから、新しいアーキテクチャとして注目されている技術である。P2P アーキテクチャではサーバやクライアントといった役割の区別がなく、それぞれが対等であるという意味を込めて、ネットワークに接続しているノードをピアと呼ぶ。P2P システムでは、複雑な設定をあらかじめ行うことなく、ピアが互いに通信することで連携しあって動作する。有名なシステムとして Napster や Gnutella などがあげられる。

このような特徴を持つ P2P アーキテクチャは、無線アドホックネットワーク上で、サーバを用いず、その場に存在する端末だけで構成されるようなシステムには有力なアーキテクチャとなる。実際に無線アドホックネットワークと P2P アーキテクチャを組み合わせたシステムに関する研究は MANET^[3]、MID Net^[4]など多く見られる。しかし、これらは P2P アーキテクチャにより無線アドホックネットワークの通信範囲を広げることには焦点を当てており、端末間の連携を対象とした研究はあまり見られない。そこで本稿では、無線アドホックネットワーク環境において、P2P アーキテクチャを採用した、端末間の協調的な意思決定の仕組みである Noroshi アーキテクチャを提案する。

2. 課題の洗い出し

本章では、無線アドホックネットワーク下で端末間の協調的な意思決定を行うときの課題を洗い出す。なお、本稿で議論の対象としている通信には、世界的に標準が確立しており、さまざまな機器にあらかじめ組み込まれている可能性の高い、インターネットプロトコル (IP) の利用を前提としている。

2.1 無線アドホックネットワーク環境における課題

ここでは、無線アドホックネットワーク環境で解決しなければならない課題を挙げる。

不安定な通信品質を解決する仕組み

通信ケーブルを使わずに電波を用いて通信を行うため、無線通信インタフェースの種類や電源状態によっては、相手ノードへはメッセージが到達するが、逆に相手ノードの電波がこちらに届かないということが起きる可能性がある。つまり、通信路が単方向性になることがあり、通信しているノード間で対等なやり取りが行えない可能性がある。また、外部環境から発生する電磁波ノイズによって通信品質が乱され、通信データが欠落したり、破損したりすることがある。また、電波の届く範囲外に端末が移動して通信不能な状態に陥ったりすることがある。そのため、通信データの欠落が起きたり、通信路が単方向性になったりしても、安定した通信を実現させるための仕組みが必要となる。

通信したい相手のアドレスを取得するための仕組み

通信を行うには、一般的に、自ノードのアドレスと、通信したい相手ノードのアドレスが必要である。しかし、その場に存在する機器だけで構成される無線アドホックネットワーク環境では、人間が手入力するなどの方法をとらない限り、通信したい相手ノードのアドレスを知ることができない。そこで、即時的な相互接続を実現するために、通信相手のアドレスをシステ

ムで検知するための手段が必要である。

直接電波の届かないノードと通信するための仕組み

比較的普及している無線通信規格である IEEE 802.11 b, 同 802.11 a, Bluetoothなどは、いずれも通信距離は数メートルから数十メートルという短い通信可能距離のため、電波が直接届く範囲から外れてしまうと、通信相手のアドレスが分かっているにもかかわらず、遠方のノードとは通信ができない。これでは、無線ネットワーク接続可能な機器が広範囲に存在しており、それらをまとめてひとつのシステムとして連携させようとした場合に、機器の存在範囲の両端に配置されている機器間では、互いに通信ができないような状況に陥ってしまう。そのため、直接無線通信ができなくても、遠方のノードとの通信が可能となるような仕組みが必要である。

2.2 協調的な意思決定を実現するための課題

無線アドホックネットワーク環境下で、複数の端末間で協調的に意思決定をおこなうための課題を解説する。

情報の収集・判断・結果の通知の仕組み

複数のノードが協調的に意思決定を行うには、他のノードがどのような状態であるのかを収集し、収集した情報に基づいて判断を行う必要がある。そして、判断を行った結果を各ノードに伝えることで複数のノードが一貫した動作を行うことができる。つまり、複数のノードが協調して動作するには、1) 情報の収集, 2) 判断, 3) 結果の通知, が必要になる。すべての情報を一箇所に集約して管理しているサーバが存在すれば、意思決定に必要な情報の収集, 判断, 結果の通知は、すべてサーバ内で完結する。しかし、無線アドホックネットワーク下での、サーバの存在しない環境のもとでは、ネットワークに接続している各ノードの情報は、それぞれのノード内にのみ存在する。したがって、意思決定を行う際には、各ノードから必要な情報をネットワーク経由で取得し、すべての情報が集まったところで判断を行い、結果を再び各端末へ送り返すという手順が必要となる。この手順を無線アドホックネットワーク環境で効率的に実行できる仕組みが課題となる。

3. Noroshi アーキテクチャの提案

3.1 Noroshi アーキテクチャの概要

Noroshi アーキテクチャとは、分散スケジュール管理システム「Noroshi スケジューラ」の基盤として開発された、無線アドホックネットワーク環境下での協調的な意思決定を実現するためのものである。サーバを必要とせず、端末間で連携することで全体のアプリケーション機能を実現する P2P アーキテクチャは、前章で洗い出された無線アドホックネットワークにおける課題を解決したり、協調的な意思決定をおこなうための情報の集約・判断・通知を効率よく実現するために最適であるという判断から、われわれは、P2P アーキテクチャを Noroshi アーキテクチャの基本的な考え方として採用した。

3.2 ピュア型 P2P アーキテクチャの採用

P2P アーキテクチャ下では、扱う情報のほとんどが分散管理されているため、アプリケーションは、必要な情報がどこに存在しているのかを簡単には把握できない。P2P アーキテクチャは、情報の発見をどのように扱っているかという観点から、ハイブリッド型 P2P とピ

ユア型 P2P の 2 種類に分類される。Napster などのアプリケーションに代表されるハイブリッド型 P2P は、各ピアがどのような情報を持っているのかというインデックス情報を管理するサーバを設置して、ピアがそのサーバに問い合わせることで、自分が必要とする情報の発見を行う。それに対し、ピア型 P2P は特定の管理サーバを持たないため、必要な情報は周囲のピアに問い合わせることで情報の発見を行うことになる。図 1 に発見のための通信と情報交換のための通信がアーキテクチャごとに異なっている様子を示す。

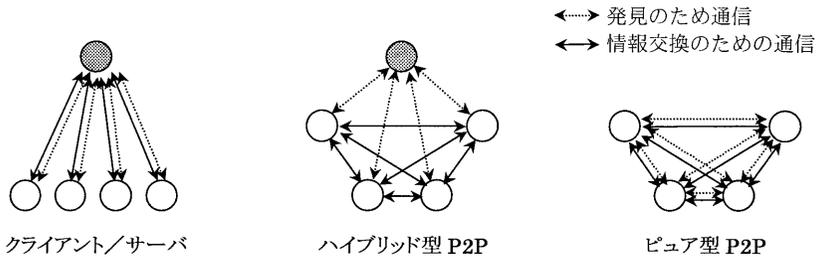


図 1 アーキテクチャごとの通信相手の違い

このように、サーバが存在しない無線アドホックネットワーク環境ではピア型 P2P アーキテクチャが適しているといえる。そこで、Noroshi アーキテクチャではピア型の P2P アーキテクチャを採用した。

3.3 通信したい相手のアドレスを取得するための仕組み

特定のサーバが存在しない無線アドホックネットワークでは、他の端末を探索することは、ピア型の P2P アーキテクチャで他のピアを探索することと同様であると考えられる。ピア型の P2P ネットワークでは各ノードが互いの存在を検知するまで、他のノードとは通信できない。そこで、Noroshi アーキテクチャでは、まず、ネットワーク上にどんなノードが存在するのかを検知するために、近隣のノードを探索することから始める。そして、さらにその近隣ノードに対して、他のノードの探索を依頼することにより、通信できるノードを芋づる式に探索していく。近隣のノードとは無線通信が直接行える範囲にいるノードのことである。このような手順により P2P ネットワークが動的に構築されていく。P2P ネットワークは定期的にノードの存在確認と新規ノードの探索を繰り返すことで、常に新しい状態を維持している。

ノード探索は、WHERE というノード探索メッセージと HERE という情報通知メッセージを用いて実行される。WHERE には探索したいノードの条件が記述されており、その条件に合致したノードが HERE を返信するというのが基本的な流れである。HERE には自ノードのアドレスが格納されている。WHERE が該当のノードに到達し、そのノードから HERE を受け取ることができれば、送信経路と受信経路が確定し、そのノードとのメッセージのやり取りが可能となる。このとき、通信経路の途中に単方向性の通信路がある場合があるので、送信経路と受信経路が異なる可能性がある。

探索したいノードが常に直接通信できる範囲に存在するとは限らないため、WHERE を受信したノードは、より遠くのノードにも WHERE を転送することで、探索範囲を広げている。このとき、WHERE が無限に転送されないように、転送回数を限定する値 (TTL 値) をメッ

セージ内に組み込み、転送されるたびに減少されるようにしている。

ノード探索の流れ

1) 自ノードの手順

- ① WHERE をブロードキャストする (図 2 ①)。
- ② HERE 作成者 (探索条件を満たすノード) から HERE を初めて受信したら、HERE 送信者 (HERE を転送してきたノード) に対して自ノードの HERE を送信する。

2) 他ノードの手順

- ① WHERE を受信した場合、以下のうち当てはまるものを実行
 - ・ WHERE 作成者 (探索要求を出したノード) に対するメッセージ送信経路を既知っているのであれば、HERE を WHERE 作成者に送信する (図 2 ②)。
 - ・ WHERE 作成者への送信経路が未知であれば、HERE を WHERE 送信者 (WHERE を転送したノード) に送信する (図 2 ③)。
- ② TTL 値を 1 減少させた上で TTL 値が 0 以上であれば、受信した WHERE をブロードキャストする (図 2 ④)。

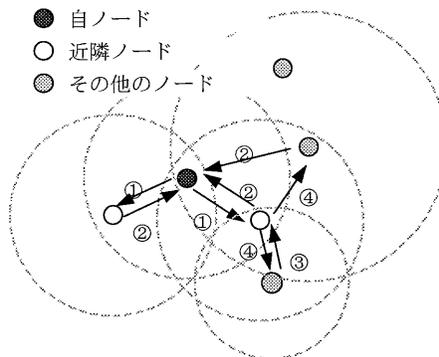


図 2 ノード探索の例

3.4 直接電波の届かないノードと通信するための仕組み

無線通信のアクセス範囲が限られていて、直接的に通信ができないノードに対してメッセージを送りたい場合、Noroshi アーキテクチャでは、近隣のノードのうち、送信先ノードへの送信方法を知っているノードにメッセージを中継してもらうことで、最終的な送信先ノードに届けられるようにしている。このような仕組みは、一般的にはマルチホップ通信と呼ばれている (図 3)。

各ノードは近隣のノードとのみ通信できればよく、最終的な送信先ノードへの送信経路を知らなくても、送信先ノードへの送信経路を知っている近隣ノードに送信することで、連鎖的に送信先ノードへとメッセージが中継される。マルチホップの実現には前述のノード探索の結果を利用し、メッセージ送信時点での最新の転送情報を参照しながら転送されていく。

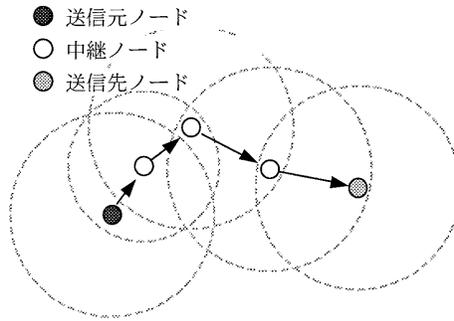


図 3 マルチホップ通信

3.5 不安定な通信品質を解決する仕組み

電波の乱れや、通信範囲内からの離脱によって通信パケットが消失する可能性がある通信路では、ノード間で通信コネクションを張る TCP/IP を用いると、通信が途切れるたびにコネクション維持やメッセージの再送のために通信パケットが大量に発生してしまうといった問題がある。そこで、Noroshi アーキテクチャは、コネクションレス型の UDP/IP を用い不用意にパケットが送信されないようにしている。しかし、UDP/IP を用いると、送信したメッセージの到達保障がおこなわれないため、アプリケーションレベルでのメッセージの到達確認と再送制御を実現しなければならない。Noroshi アーキテクチャではメッセージの送受信に ACK を用いた到達確認と再送制御を組み込むことにより安定性を高めている。ところが、単方向性の通信路においてはメッセージを送信しても ACK が戻ってこないために、相手に到達したかの確認が取れず、何度も再送してしまうことになる。これでは送信先に全く同内容のメッセージが重複して届いて誤作動の原因になりかねないため、生成された各メッセージに対して固有の ID を割り当て、メッセージを受信した際に一定期間は ID を保持しておくことで、以後の同一メッセージを重複して受信しないような制御を取り入れた。相手ノードが無線通信できない場合は、すぐにメッセージを送信できないので、自ノード内に用意されたメッセージ・キューに送信に失敗したメッセージをしばらく保留しておき、定期的に送信を試みる。

3.6 情報の収集・判断・結果の通知の仕組み

協調的に意思決定を行うためには、情報の収集・判断・結果の通知という手順を、無線アドホックネットワーク環境下で効率よく実行する必要がある。P2P アーキテクチャでは、クライアント/サーバアーキテクチャでのサーバのように、情報が集約されるノードが常に存在するとは限らない。しかし、全体的な判断を行うには少なくとも一つのノードにすべての情報が集約されなければ判断を下すことができない。そこで、情報を収集するという視点から、通信メッセージの交換方法を以下のように検討した。

1) 情報集約型

意思決定を行いたいノード（主催者ノード）が、他の全てのノードに、情報を問い合わせるためのメッセージを送信する。メッセージを受けたノードは主催者ノードに情報を送り返す。全ての情報を取得することができるのは、主催者ノードのみである。したがって、「判断」を行うことができるのは主催者ノードのみである（図 4 情報集約型）。

2) 情報交換型

意思決定を行いたいノード（主催者ノード）が、他の全てのノードに、情報を問い合わせるためのメッセージを送信する．そのメッセージには「判断」するための詳細な条件を付加しておく．問い合わせ要求を受けた各々のノードは「判断」に必要な情報を、他のすべてのノードに対して送信する．全員が互いに全員の情報を知ることができるため、誰でも「判断」を行うことができるという利点がある（図4 情報交換型）．

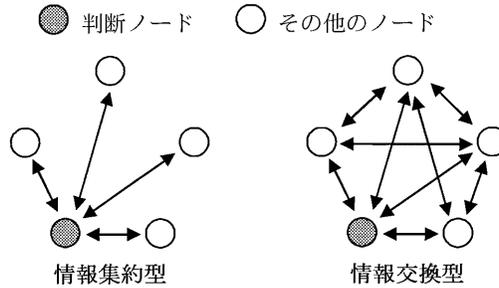


図4 メッセージ通信方式の比較

1) の情報集約型では、主催者ノードへの負荷の集中が懸念されるほか、無線アドホックネットワークにおいては主催者が無線範囲内から外れると、いつまでも「判断」が行えないという問題点がある．また、2) の情報交換型では、メッセージの流量が膨大になってしまい、電波大域を共有する無線ネットワークでは好ましくない．また、全員が持っている情報が必ず同等であるという保障も無いため、判断結果もノードによって異なる可能性がある．

回覧型のメッセージ通信

そこで、新たに回覧型のメッセージ通信方式を考案した（図5）．各ノードでメッセージが回覧されていく過程で、各々のノードが「判断」に必要な情報をメッセージに書き込んでいくことで、最後にメッセージを回覧したノードはすべての情報を把握できる．そして、そのノードが判断を下し、判断結果を再び回覧することで、全てのノードに判断結果を通知することができる．

回覧型のメッセージ通信では、情報の流れを1本にすることで特定のノードに通信負荷がかかることも無く、また、一旦メッセージを送ってしまえば、あとは結果を待つだけでよいので、電源を切ってしまったたり、電波範囲から外れてしまったたりしても、全体の判断が遅れる可能性が小さい．

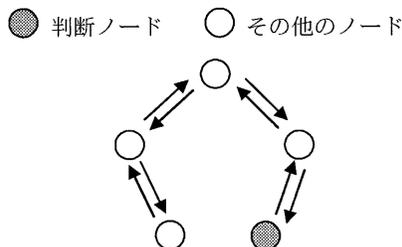


図5 回覧型のメッセージ通信

われわれは、この回覧型のメッセージを一筆書きメッセージと呼び、往復型の ROUND_TRIP, 周回型の GO_AROUND, 一方通行型の ONEWAY を考案した。いずれのメッセージについても、配送のための情報として、回覧するノードのリストと回覧済みフラグを持っており、リストの順番に沿ってメッセージが配送されていく。メッセージを受信したノードはメッセージの回覧済みフラグをチェックし、次のノードに送信する。途中オフラインのノードが存在した場合は、そのノードを後回しにして回覧を続けていく。

一筆書きメッセージのうち、協調的な意思決定に用いる ROUND_TRIP は、指定されたノードのうち、最後に到達したノードを折り返し地点として、往路、復路とも全員が回覧しながら戻ってくるメッセージである。往路で情報を収集し、折り返し地点で「判断」をおこない、復路で決定情報を通知する(図6)。

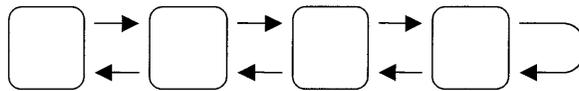


図6 一筆書きメッセージ (ROUND_TRIP)

メッセージへの判断条件の添付

一筆書きメッセージの折り返し点で「判断」を行うために、判断条件自体を ROUND_TRIP メッセージの内容の一部として添付する。これを判断ポリシーと呼ぶ。これにより、「判断」をおこなうノード(つまり折り返しノード)が誰になっても、確実に判断条件を得られるように、判断ポリシーの内容は、ROUND_TRIP メッセージの往路で収集した情報に基づいて、アプリケーションが判断を実施するために必要な情報を表現したものである。

判断の前倒し

判断ポリシーをメッセージに添付することで、メッセージの回覧途中であっても、その時点で収集されている情報の範囲内で判断を下すことが可能な場合もある。その場合は、早期に判断を行うことで、一筆書きメッセージを直ちに折り返して結果を通知することが可能になり、無駄なメッセージを送信する必要がなくなる。

4. Noroshi アーキテクチャの適用例

4.1 Noroshi アーキテクチャの概要

本章では、Noroshi アーキテクチャを利用した分散スケジュール管理システム「Noroshi スケジューラ」において、協調的な意思決定によって複数の情報端末間で共通の予定を新規に作成するという例を示す。Noroshi スケジューラは、Microsoft PocketPC 環境で動作するアプリケーションとして実装されており、グループによるノードの管理機能を備え、グループ内の暗号化通信をサポートし、アドホックネットワーク上のセキュリティにも配慮している。また、PocketPC に標準で組み込まれている Pocket Outlook と予定のデータを共有できるため、デスクトップ PC との予定の同期も可能である。動作画面を図7に示す。



図7 Noroshi スケジューラ
動作画面

4.2 スケジュール調整への適用

Noroshi スケジューラ上で、予定の新規作成要求を出すと、予定の成立条件が添付された ROUND_TRIP メッセージが、予定の参加者に順番に送信される。その過程で、スケジュールの空き時間を収集しながら、条件を満たすような予定が成立するか否かを判断する。

Noroshi におけるスケジュール調整のためのメッセージ

Noroshi スケジューラは予定を新規に作成するためのメッセージ (CREATE メッセージ) を表 1 のように定めている。CREATE メッセージは状態を持っており、その状態は ROUND_TRIP の回覧の行程の中で変化する。状態によってノードが行う処理が異なる。

表 1 予定新規作成メッセージ CREATE の詳細

メッセージ状態	意味	このメッセージを受信したときのノードの処理
PREPARE	予定が未確定である	予定として指定された時間帯を予約し、判断ポリシーを評価する。評価の結果、予定が成立することが確定すれば COMMIT 状態に遷移する。逆に成立しないことが確定したら ROLLBACK 状態に遷移し、ROUND_TRIP を直ちに折り返す。
COMMIT	予定の成立が確定している	予定として指定された時間帯が空いていれば確保し、予定を登録する。空いてなければ直ちに次のノードへ転送する。
ROLLBACK	予定の不成立が確定している	PREPARE 状態のときに予約してある時間帯を解放し、次のノードに転送する。

Noroshi における判断条件

Noroshi スケジューラでは判断ポリシーとして以下のようなものを作成した。これらは複合的に指定することができる。

- 調整期限 : スケジュールの調整期限である。スケジュールの調整が完了せずに調整期限を過ぎた場合は、調整中のスケジュールは自動的に破棄される。
- 最低参加人数 : スケジュールに参加すべき最低の参加人数である。指定した人数が確保できた場合にスケジュールが成立する。
- 必須ノード : スケジュールに参加すべき特定のノードである。指定されたノードが全員参加できる場合にスケジュールが成立する。

スケジュール新規調整の例 (ケース 1)

A は主催者として、これから A,B,C,D を参加者とした新しい予定を調整しようとしている。判断ポリシーとして「最低参加人数が 3 人以上」が指定されており、C,D には空き時間がない (図 8)。

- ① A が作成した CREATE メッセージは PREPARE 状態であり、最初に A 自身が CREATE の処理を行い、指定された時間帯を予約する。この時点で判断ポリシーを評価し、いまだ 3 人の参加者を得ていないことが分かるので、PREPARE 状態のまま、次の参加者に転送する。
- ② B は A から PREPARE 状態の CREATE を受信したため、該当の時間を予約して判断ポリシーを評価する。いまだ 3 人の参加者を得ていないので、PREPARE 状態のまま、次の参加者に転送する。
- ③ C は B から PREPARE 状態の CREATE を受信したが、該当の時間に空きがないた

め、予約できない。その後、判断ポリシーを評価するが、いまだ3人の参加者を得ていないので、PREPARE 状態のまま D に送信する。しかし D とは通信不可能なため、メッセージを D への送信キューに登録し、D と通信可能になるまで保持する。

- ④ D が通信可能になったため、C からメッセージを受け取るが、D にも空き時間が無い。ここで判断ポリシーを評価するが、既に全員の回覧が完了しているにも関わらず参加者が3人に満たないので、予定が不成立であることが確定する。したがって CREATE を PREPARE 状態から ROLLBACK 状態に変更して、C に送り返す。
- ⑤ C は ROLLBACK 状態の CREATE を受信したため、予約されている時間帯を解放して、B,A と復路に対してメッセージ送信する

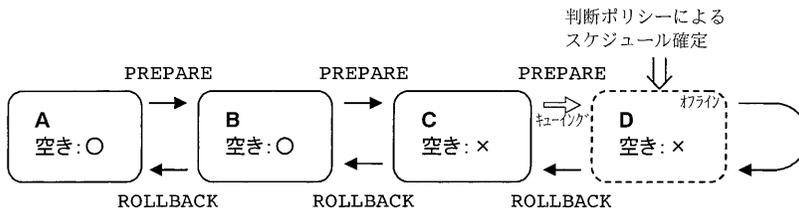


図 8 「最低参加人数が3人」というポリシーに基づいたスケジュール新規

スケジュール新規調整の例 (ケース2)

A は主催者として、これから A,B,C,D を参加者とした新しい予定を調整しようとしている。判断ポリシーとして「必須参加者が B で3人以上参加」が指定されており、B には空き時間がない (図9)。

- ① A が作成した CREATE メッセージは PREPARE 状態である。最初に A 自身が CREATE の処理を行い、指定された時間帯を予約する。この時点で判断ポリシーを評価し、いまだ B の参加の有無も3人の参加者も得ていないので、PREPARE 状態のまま、次の参加者に転送する。
- ② B は A から PREPARE 状態の CREATE を受信したが、該当の時間に空きがない。これこれ判断ポリシーを評価するが、必須の参加となっている B が出席できないことが分かるので、予定が不成立であることが確定する。したがって CREATE を PREPARE 状態から ROLLBACK 状態に変更して、A に送り返す。
- ③ A は ROLLBACK 状態の CREATE を受信したため、予約されている時間帯を解放する

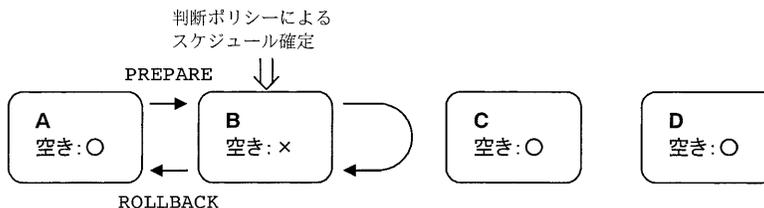


図 9 「必須参加者が B で3人以上参加」というポリシーに基づいたスケジュール新規

5. おわりに

本稿では、無線アドホックネットワーク環境下で情報端末間の協調的な意思決定を支援する Noroshi アーキテクチャの提案を行った。Noroshi アーキテクチャでは、不安定な無線環境下でも、安定して通信できるような仕組みを取り入れたほか、P2P アーキテクチャを採用することで、無線アドホックネットワーク下でのさまざまな課題を、端末間で連携することにより解決できた。協調的な意思決定を実現するメッセージ通信機構は、一筆書きのように、メッセージが配送される経路を1本にし、判断条件を同時に送信することで、意思決定に関するすべての情報を収集することなく、判断可能な量の情報が集まった時点で矛盾無く結論が出せるようになった。また、送信できる相手からメッセージを配送することにより、耐障害性も高めることができた。

ユビキタスコンピューティング時代への流れとともに、いままでは主にコンピュータが接続していたネットワークにも、センサデバイスや情報家電などの非コンピュータ機器が無線により大量に接続してくることが予想される。本稿ではこうした環境に対する基本的な問題とその解決例を提示したが、さまざまなデバイスをITシステムに統合できるようにするため、今後も新しいシステムアーキテクチャを模索していく必要があるだろう。

* 1 Noroshi の研究開発は、通信・放送機構の委託研究として、独立行政法人産業技術総合研究所サイバーアシスト研究センターと共同で行われたものである。

- 参考文献** [1] <http://webs.cs.berkeley.edu/>
[2] http://www.xbow.com/Products/Wireless_Sensor_Networks.htm
[3] http://www.ietf.org/html.charters/manet_charter.html
[4] <http://www.acr.atr.co.jp/acr/general/event/workshop/pdf/workshop2.pdf>

執筆者紹介 渡 邊 充 隆 (Mitsutaka WATANABE)

2000年名古屋大学工学部電気電子情報工学科卒業。同年日本ユニシス(株)入社。対外系金融システムの開発、通信・放送機構の委託研究での Noroshi 開発を経て、現在はRFIDを用いたビジネス開発に従事。サービスビジネス統括部 ユビキタスビジネス開発部に所属。