

## ロボット制御のための P2P 通信システムの設計と実装

緒方 雄一<sup>†1</sup> スパホ エブヨラ<sup>†1</sup>  
松尾 慶太<sup>†2</sup> バロリ レオナルド<sup>†3</sup>

近年 P2P の研究は、動画配信技術や、情報分散処理への応用が活発に行われている。P2P を応用することでこれまでに無いサービスを提供することができる。P2P 技術はクライアント/サーバ技術の通信の境界を越え、ユビキタス社会の構成に大きく関与している。ロボット技術は近年のデジタル技術の普及に伴い急速に発展してきた。しかしインターネット側から Firewalls や NAT (Network Address Translation) をクライアント/サーバで超えて装置を制御することは困難であり、P2P 技術の方が適している。そこで我々は新しく P2P 技術とロボット制御を組み合わせた JXTA-Overlay P2P 制御システムを提案、実装した。また、JXTA-Overlay P2P 制御システムを用いて実験を行い、ロボットを制御するために評価を行った。その結果、提案システムはロボット制御に適していることがわかった。

### Design and Implementation of P2P Communication System for Robot Control

YUICHI OGATA,<sup>†1</sup> EVJOLA SPAHO,<sup>†1</sup> KEITA MATSUO<sup>†2</sup>  
and LEONARD BAROLLI<sup>†3</sup>

The design of an efficient collaborative multi-robot framework that ensures the autonomy and the individual requirements of the involved robots is a very challenging task. This requires designing an efficient platform for inter-robot communication. P2P is a good approach to achieve this goal. P2P aims at making the communication ubiquitous thereby crossing the communication boundary and has many attractive features to use it as a platform for collaborative multi-robot environments. In this work, we presented the JXTA-Overlay P2P system and its application for robot control. Since JXTA-Overlay is able to overcome Firewalls, Routers and NATs, it is possible to control end-devices in a WAN without changing the network security policy. We used JXTA-Overlay for the control of robot motors. We evaluated the proposed system by many experiments and have shown that the proposed system has a good performance and can be used successfully for the control of robot.

### 1. はじめに

近年、家庭用 PC の性能向上に伴いクライアント/サーバシステムとともに、P2P (Peer to Peer) システムが注目され、多くの研究が行われている。P2P 技術の多くは Napster<sup>1)</sup> や Winny などのファイル共有や、Windows Live Messenger<sup>2)</sup> や Skype などの PC 対 PC ネットワークに用いられている。P2P 技術を用いることにより端末数が膨大になっても通信が可能という利点がある。しかし P2P 技術を用いたソフトの多くはソースコードが非公開である。

一方、ロボット技術では、近年のデジタル技術の普及に伴い急速に発展してきた。現在、ロボットは自動車のパーツ組み立てや食品の生産など、単純作業を行うロボット、超温度差、真空など人が生存できない場所で代わりに作業するロボット、などが活用されている。さらにより人間を意識したロボットの研究も盛んに行われており、体に装着して身体機能を拡張、増幅するロボット<sup>3)</sup>、などの非常に高い技術の報告があげられている。これからの時代、ロボットを制御するにあたり、多くの情報を共有し制御することが求められる。

Sun Microsystems が提唱する JXTA<sup>4)</sup> は現在の TCP/IP ベースの限界を超えて、インターネットの及ぶ範囲を拡大することを目的に開発された P2P プラットフォームであり、オープンソースとして公開されている。JXTA は、現在多くの研究が行われており、おもに、インターネット上にある様々な資源を P2P システムで共有し、有効に活用しようとする研究が中心である。例えば、世界中に点在する Peer(端末)を組織的にまとめあげる Grid Application(並列分散システム)<sup>5)</sup> としての応用、P2P を運用する際のインターネット上におけるアドレスの処理に関する研究、P2P システム特有のオーバーレイネットワークを応用した E-learning 教材の提案<sup>6)</sup> などがあげられる。

これまでの研究では、クライアント/サーバを用いた遠隔ロボット制御が主流であった。本研究では、P2P 技術を考慮し新たなシステムの提案、実装を行った。また、提案シス

<sup>†1</sup> 福岡工業大学大学院工学研究科

Graduate School of Engineering, Fukuoka Institute of Technology (FIT)

<sup>†2</sup> 福岡県立嘉穂総合高等学校

Fukuoka Prefectural Kaho-Sogo High School

<sup>†3</sup> 福岡工業大学情報工学部

Department of Information and Communication Engineering,

Fukuoka Institute of Technology (FIT)

ムがロボット制御を行うのに適しているか、評価を行った。

以下、第2節、第3節では、研究に用いたAPIについて述べる。第4節、第5節では今回新たに実装した Robot Control System (RCS) と、JXTA-Overlay P2P 制御システムについて述べる。そして、第6節で新システムの評価を行い、最後に第7節にて結論と今後の課題を述べる。

## 2. JXTA プラットフォーム

JXTA は、その設計にあたって、できる限り多様な P2P アプリケーションのニーズに対応できるように配慮されたプラットフォームである。また、様々な種類の P2P アプリケーションの基礎となるコア P2P 機能の設計を目標として開発された。JXTA は次のサービスを Peer へ提供する。

- 他の Peer とそのサービスを発見する。
- 自分が提供できるサービスを公開する。
- 他の Peer とデータを交換する。
- メッセージを他の Peer にルーティングする。
- Peer を集団化して Peer グループを形成する。

JXTA がプラットフォームとして定義する一連のプロトコルは、Peer がオペレーティングシステムや開発言語、ネットワークトランスポートなどに左右されることなく、堅牢で汎用なネットワークを形成するために必要な機能を実装している。

### 2.1 JXTA コアの設計理念

JXTA コアを設計するにあたり、最も重要視されたことはオペレーティングシステムや開発言語に依存しないということであった。その理由は、どのような P2P ネットワーキングアプリケーションでも JXTA プロトコルを使用できることを目指し JXTA プロトコルを利用するアプリケーション開発者を拡大したいという意図があったためである。さらに、JXTA プラットフォームは、ネットワーク Peer として組み込みデバイスからスーパーコンピュータクラスにいたるまで、様々な種類のデバイスを想定して設計されている。また、JXTA が乗り越えられたのは、オペレーティングシステムやコンピュータプラットフォームやプログラムの開発言語の種類を問われないことだけではない。JXTA は、ブロードキャストやマルチキャストを要求してはならないことを除けばネットワークトランスポートのいかんも問わない。

加えて、JXTA は次の二つを仮定している。

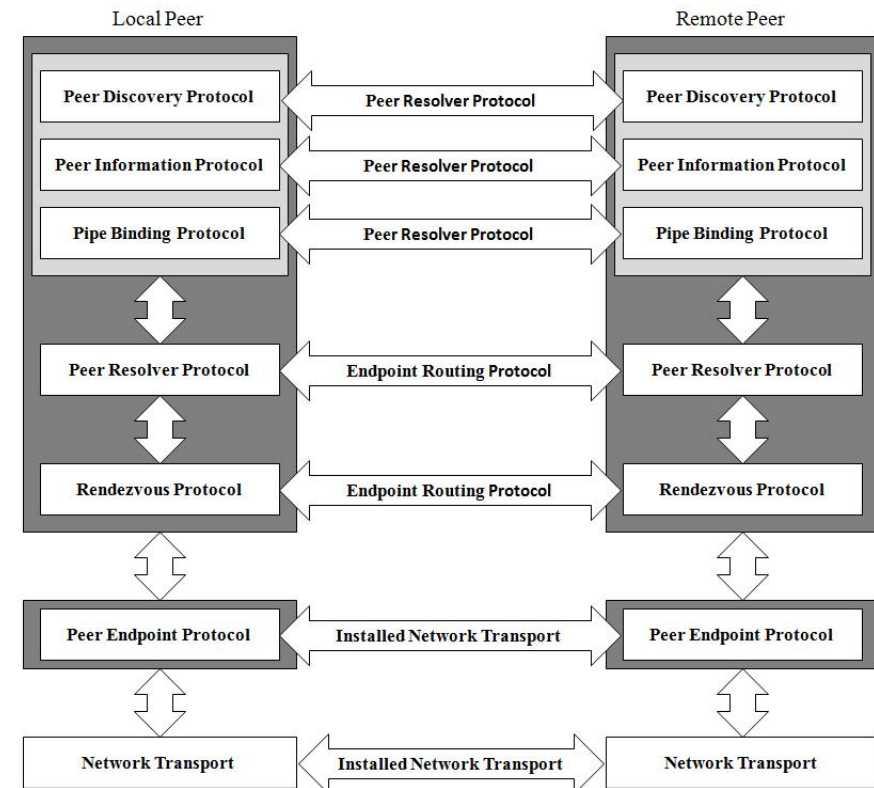


図 1 JXTA プロトコルスタック

- (1) Peer とそのリソースは、ネットワーク上に現れたり消えたりする可能性がある。
- (2) Peer は、そのネットワーク上での位置が変わったり NAT やファイアウォールによりプライベートネットワーク内に隠されることがある。

さらに、JXTA は、Peer が情報をキャッシュしてネットワークトラフィックを減らすことと、ネットワークに直接接続していない Peer へのメッセージルーティングを提供することを推奨している。

### 2.2 JXTA プロトコル群

JXTA は、XML メッセージベースで次の6つのプロトコルで設計されている。

- Peer Discovery Protocol
- Peer Resolver Protocol
- Rendezvous Protocol
- Peer Information Protocol
- Pipe Binding Protocol
- Endpoint Routing Protocol

これらの JXTA プロトコルは、一つ一つが P2P ネットワーキングの別々の問題に取り組むようになっている。また、プロトコル会話は、ローカル Peer によって行われる部分と、リモート Peer によって行われる部分の二つに分けられる。ローカル Peer の役目は、メッセージを生成してリモート Peer へ送ることである。リモート Peer の役目は、メッセージを受け取り、それを処理してタスクを実行することである。また、それぞれの JXTA プロトコルは他のプロトコルと半独立の関係にある。これにより、Peer は一部のプロトコルだけを実装して機能を提供し、事前に定めた動作に従うことによって一部のプロトコルを省くことができる。例えば、予め構成しておいた一群のルータ Peer に頼ってルーティングを行う Peer は、Endpoint Routing Protocol を実装する必要はない。しかし、JXTA プロトコルは他の JXTA プロトコルと完全に独立しているわけではない。JXTA プロトコルスタック (図 1) の各レイヤーは他の Peer へ接続を提供する上で、すぐ下のレイヤーに依存している。Peer Discovery Protocol の独立した実装を構築することは可能であっても、Peer Resolver Protocol と Endpoint Routing Protocol の実装なしに、リモート Peer へのメッセージのトランスポートを処理することはできない。このように、JXTA プロトコルは、一部のプロトコルだけの実装もできる形をとってはいるが実際にはプロトコル全体を完全に実装することを推奨している。

### 2.3 JXTA の論理レイヤー

JXTA の各レイヤーは図 2 のように直下のレイヤーを土台として動作している。

コアレイヤーは、いかなる P2P ソリューションにおいても不可欠な以下の要素を提供するレイヤーである。

- Peer と Peer グループ
- ネットワークトランスポート (パイプ, エンドポイント, メッセージ)
- アドバタイズメント (広告)
- エンティティネーミング (識別子)
- プロトコル (発見, 通信, 監視)

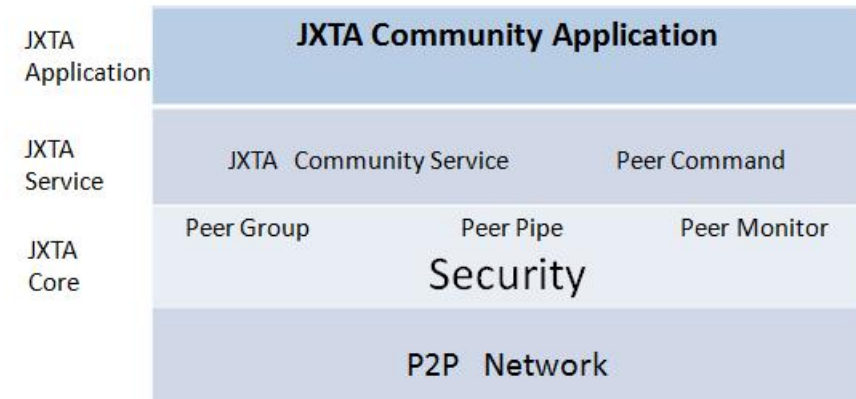


図 2 JXTA の論理レイヤー

- セキュリティと認証

コアレイヤーには JXTA が提供する六つのメインプロトコルが含まれる。このレイヤーは、JXTA ソリューションの中核をなすものである。図 2 に示すように JXTA のサービスレイヤーやアプリケーションレイヤーはこのレイヤーを土台として機能を提供している。

### 3. JXTA-Overlay

JXTA-Overlay は JXTA プロトコル上で JXTA ベースアプリケーションに必要とされる基本的な機能を提供するものである。JXTA-Overlay は次のような機能を提供している。

- Peer discovery
- Peer's resources discovery
- Resource allocation
- Task submission and execution
- File/data sharing, discovery and transmission
- Instant communication
- Peer group functionalities
- Monitoring of peers, groups, tasks etc.

ここに示すものは、JXTA ベースのアプリケーション開発時に必要となる基本的な機能で

ある。これらの機能は JXTA レイヤー上に独立して構築されており、他のアプリケーションから呼び出して使用することができる。

### 3.1 Overlay の概要

ここからは、Overlay の概要について述べる。Overlay は JXTA レイヤー上に構築される P2P ネットワークである。Overlay は JXTA アーキテクチャをもとにしたものであり次の四種類の Peer を定義している。

- Minimal Peer
- Full Peer
- Rendezvous Peer
- Relay Peer

Overlay はネットワーク上に存在するすべての Peer (Client) やランデブー Peer (Broker) によって構成される。Broker は他の Peer の位置情報やファイル共有のための情報を収集する。Client は二つのタイプに分けられる。ひとつは、ユーザーインターフェイスを持つクライアント (Client) とユーザーインターフェイスを持っていないが通信に必要なリソースを提供するシンプルクライアント (Simple Client) である。

### 3.2 Overlay の内部アーキテクチャ

Overlay は 3 タイプの Peer によって構成される。図 3 に示すように、Broker, Client, Simple Client の三つである。

#### 3.2.1 Control Layer

Overlay 内部の Control Layer は大きく二つに分けられている。図 3 に示すように、上下のパートに分けられる。Control Layer の下部のパートはランデブー Peer や端末の Peer など、その種類に関係なくすべての Client に共通な機能を提供する。一方、上部パートはブローカー専用とクライアント (Client, Simple Client) 専用の二つに分けられる。

- Control Layer の下部パートは、JXTA によるメッセージ転送や Peer の発見やそのアドバタイズメント (他の Peer への告知) というような機能を提供する。
- Control Layer のブローカー用上部パートは、ブローカーのグループ管理やブローカー情報に関する統計などの維持管理を行う。
- Control Layer のクライアント用上部パートは、クライアントのグループ管理やクライアント情報に関する統計などの維持管理を行う。さらに、共有ファイルやユーザー環境の管理、ブローカーによる接続の生成を行う。

このレイヤー (Control Layer) は、JXTA ピアグループを管理し、レイヤー内部に over-

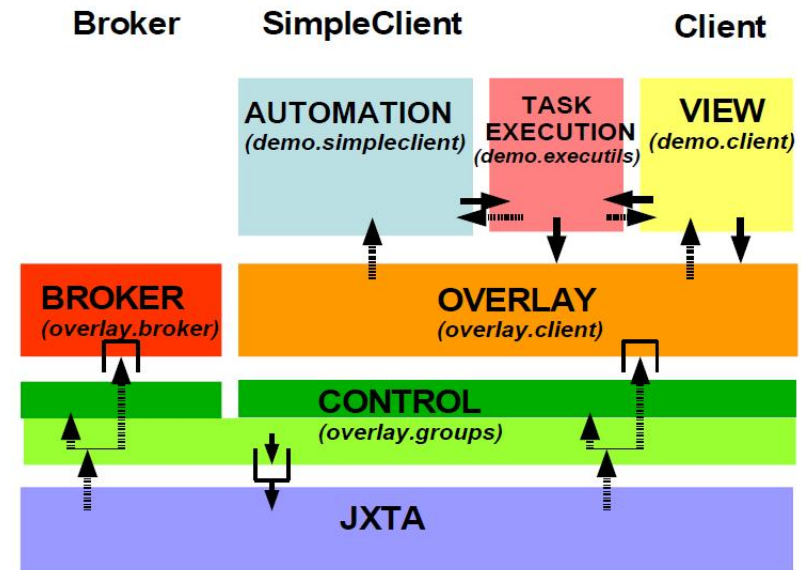


図 3 Overlay アーキテクチャ

lay.groups の名前を管理するネーム空間を持っている。さらに、このレイヤーはメッセージの送信を支援する。メッセージが届くたびに、上下のレイヤーイベントを発する。これにより、メッセージが届いたことを Overlay に伝えたり、送信元へ JXTA レイヤーを経由してメッセージが到着したことを知らせることができる。

#### 3.2.2 Business Layer

Business Layer は、Control Layer の上層レイヤーである (図 3 の BROKER と OVERLAY の層が Business Layer)。このレイヤーは、Overlay の基本機能を実装している。さらに、Broker と Client 用に特化して構成されている。そのため、二つのパッケージに分割されている。それぞれ overlay.broker と overlay.client と呼ぶ。overlay.client は、Primitives のフォームの中に Overlay API (Application Program Interface) を定義している。Primitives は、Overlay API を提供するための関数の集合体である。また、関数の実行で発生する Events をここでは Fires と呼ぶ。さらに overlay.client は、アプリケーションを開発するための強力なミドルウェアである。現在の Overlay API は次の五つのエリアに分

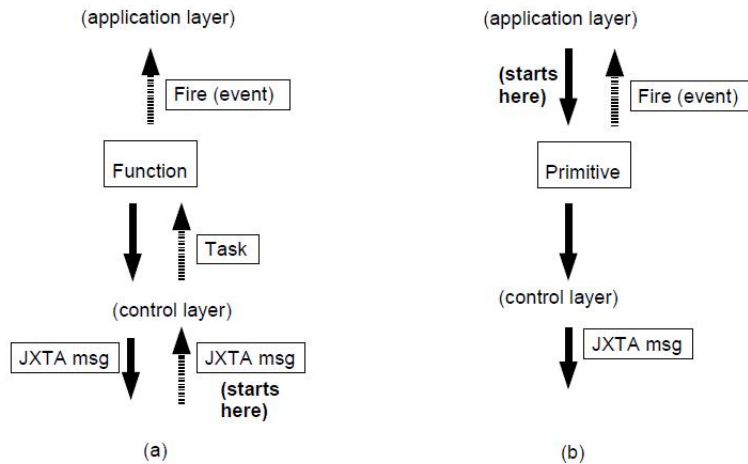


図 4 Receiving JXTA messages (a) and executing primitives (b)

けられている。

- Discovery: Network information area.
- Messenger: Instant messaging area.
- Files: File transfer area.
- Execution: Remote execution of task area.
- Learning: Groupware tools area.

図 4 に Functions, Primitives, Tasks, および Files の関係を示す。図 4 (a) では、JXTA メッセージが届いた時の処理を示している。JXTA メッセージが到着すると Control Layer は Task を Function に渡して Fire を発生させる。この Fire により Application layer はメッセージを受信したことを確認できる。さらに、送信元へメッセージを送信したことを知らせるために、確認メッセージを送信する。次に、図 4 (b) では、Application layer サイドからメッセージを送信する場合を説明している。まず、Application layer はメッセージ送信のために Overlay API を Primitive を使って呼び出す。その後、Control Layer は指定された宛先へメッセージを送信する。

次に、Business Layer の overlay.broker について述べる。overlay.broker は、先に述べた Overlay API の五つのエリアの中に統合されている。そのアーキテクチャは図 4 (a) と同

じであり、Task が実行される度に Fires を発生する。overlay.broker は、overlay.client の機能に加え次のようなサービスを追加実装している。

- ファイル検索のための協力を申し出ている Peer の発見やその処理の支援。また、その他の様々な処理を支援する。
- 各 Peer からのログインを処理し、Peer グループなどを管理する。
- Overlay データベース内の情報 (users, groups, learning tasks, etc.) にアクセスするための支援機能

### 3.2.3 Application Layer

Application Layer は、Overlay API を呼び出して使うことができる。このレイヤーは、ユーザーが JAVA で定義した複数のアプリケーションを同時に実行できることに加え、命令をバックグラウンドで動作させることができる。

## 4. Robot Control System(RCS)

RCS(Robot Control System) は、シリアル通信 (RS-232C) で接続された複数のサーボモータを制御するために新しく実装されたアプリケーションである。機能として、バーストアクセスによる一括駆動、回転位置情報の伝送とデータベースの作成、データベースを用いた連続駆動を提供する。作成されたデータベースは JXTA-Overlay P2P 制御システムに用いることも可能である。図 5 が起動画面である。我々は RS-232C 接続により駆動する Vstone 社製の三つのサーボモータと二つの基盤 (図 6) を用いて制御を行った。

### 4.1 バーストアクセス

複雑な動作をサーボモータで行うには、一斉に駆動させる必要がある。シリアル通信での通信方式は特定の一つのノードへ情報を送るユニキャスト、特定の複数のノードすべてに同一情報を送るマルチキャスト、すべてのノードへ同一情報を送るブロードキャストである。しかし、ユニキャストでは最初と最後に情報が送られてくるサーボモータの間で駆動時差が発生する。ブロードキャストの場合は異なる情報が送信できず複雑な駆動が出来ない。そこで、ユニキャストとブロードキャストを組み合わせたバーストアクセス通信方式を用いた。バーストアクセスは、図 7 に示すように二段階に分け通信を行う。

- (1) ユニキャストにて接続されたサーボモータ一つずつに回転角度情報のみを送信する。
- (2) ブロードキャストにて接続されたサーボモータすべてに駆動命令を送信する。

バーストアクセスにより、すべてのサーボモータが同時に駆動することができ、複雑な動作を実現している。

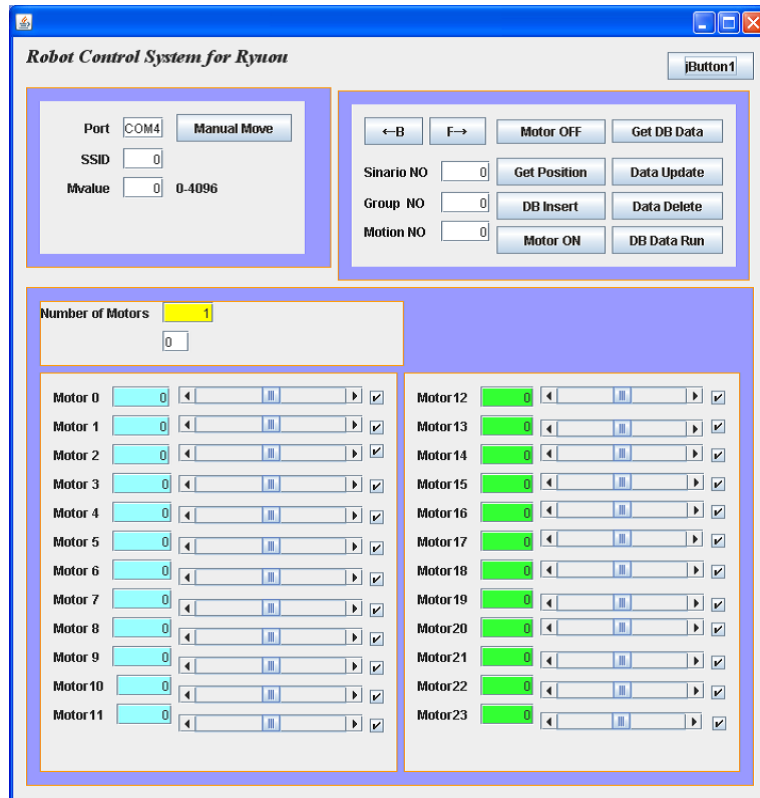
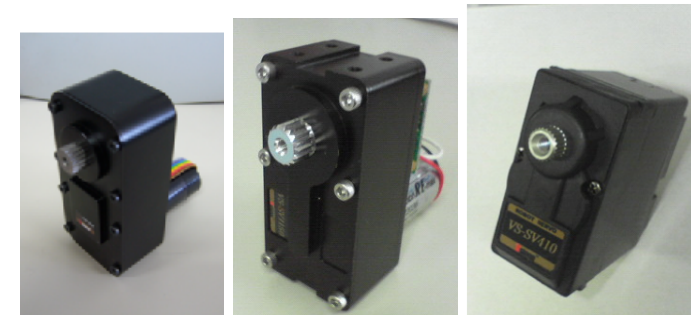


図 5 Robot Control System 起動画面

## 5. JXTA-Overlay P2P 制御システム

我々は JXTA-Overlay を用いて新たな P2P アプリケーションの提案と実装を行う。従来のファイル共有やチャット機能に加え、コマンドによりデータベースへアクセスし、シリアル通信接続されたサーボモータの遠隔制御を実装している。図 8 がサーボモータとの接続図である。以下の五段階を通してサーボモータを制御している。

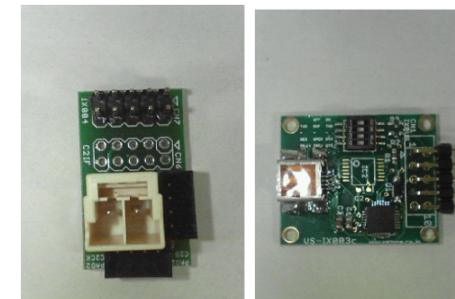
- (1) Peer A より、入力されたコマンドは Overlay Network を通じて Peer B に送る。
- (2) Peer B より、Peer B に蓄積されたデータベースを元に VS-IX003 へ送る。



(a) VS-SV3310

(b) VS-SV1150

(c) VS-SV410



(d) VS-CN005

(e) VS-IX003

図 6 Vstone 社製サーボモータと基盤

- (3) VS-IX003 より、LV serial (RS-232C) 信号に変換され、VS-CN005 へ送る。
- (4) Power supply より、電源を VS-CN005 へ供給する。
- (5) VS-CN005 より、LV serial 信号と電源をサーボモータへ送る。

## 6. 提案システムの評価

今回我々が新しく開発した JXTA-Overlay P2P 制御システムがロボットを制御する際に実用レベルかどうかを評価するために、100 回の Peer A と Peer B 間のコマンドの送受信を行い、反応速度を評価した。評価の基準として Peer B の反応速度を三つのレベルに分けた。

- 1 秒未満

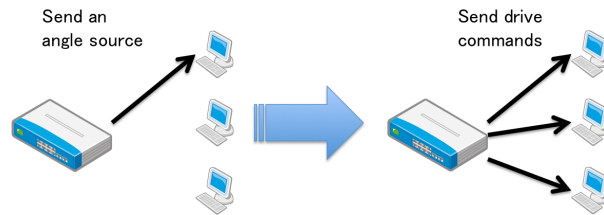


図7 バーストアクセス

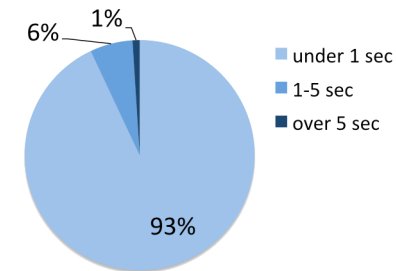


図9 反応速度による性能評価

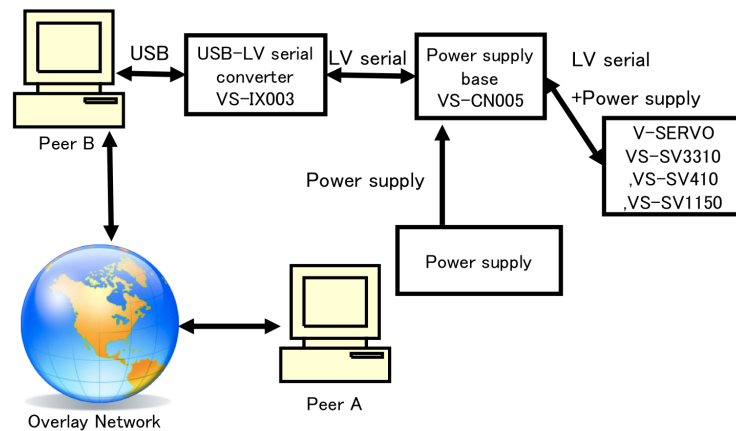


図8 提案システムによるサーボモータ接続図

- 1秒以上, 5秒未満
- 5秒以上

その結果を図9に示す。1秒未満にコマンドが実行される確率が全体の93%で、1秒以上、5秒未満が6%、5秒以上が1%。平均約0.18秒という結果になった。

精密なロボットを制御するには、情報が伝達されるまでに1秒以上必要なのは致命的である。よって本システムは、物を運ぶなど頻繁な操作が必要ない倉庫内作業ロボット制御レベルといえる。

## 7. まとめ

本システムはロボット制御のためのP2P通信システムの設計と実装を行った。まず、JXTAプラットフォームを紹介し、JXTA-Overlayについて説明した。次にRCSについてのべた。最後に、JXTA-Overlay P2Pについてのべ、ロボットを制御するために評価を行った。評価の結果、簡易作業ロボットを制御するには十分な速度であり、実用レベルであるという結論に達した。今後の課題として、精密な時間での伝達速度の測定を行うのと、測定回数を増やし、正確な評価にしていく必要がある。

## 参考文献

- 1) Napster, <http://www.napster.com/choose/index-default.html>
- 2) Windows Live Messenger, <http://messenger.live.jp/>
- 3) HAL(Hybrid Assistive Limb), <http://www.cyberdyne.jp/robotsuithal/>
- 4) Sun Microsystems, JXTA Community Projects, <https://jxta.dev.java.net/>
- 5) S. Caballe, F. Xhafa, T. Daradoumis, J. Esteve, L. Barolli and A. Durresi, "Using a Grid Platform for Enabling Real Time User Modeling in On-line Campus", Proc. of Complex, Intelligent and Software Intensive System (CISIS-2007), pp. 5-42, April 2007.
- 6) K. Matsuo, L. Barolli, F. Xhafa, A. Koyama and A. Durresi, "Implementation of a JXTA-based P2P e-Learning System and Its Performance Evaluation", Journal of Distance Education Technologies (JDET), Vol. 4, No. 3, pp. 352-371, 2008.
- 7) ICQ.com, <http://www.icq.com/>
- 8) SETI@home, <http://setiweb.ssl.berkeley.edu/index.php>