F-017

P2P型制御システムの試作

幸谷智紀* 大楯弘順*

Trial Implementation of a P2P control system

Tomonori Kouya* Kojun Ohsugi*

1. 初めに

本稿では，安価なLAN構成規格であるEthernet(物理層・ネットワーク層)と，無料かつ事実上の世界共通規格であるTCP/IPを用いた，自動修復が可能な分散制御システムの提案を行う。

このシステムは多細胞生物における再生システムのアナロジーに基づいて構築されている。生物では破壊もしくは機能に異常を来たした細胞が生じると，その近くに存在する幹細胞(stem cell)というオールマイティな細胞が，一定時間経過後に破壊された細胞と同じ役割を果たすよう変化していく。本システムにおいても，I/Oノード(以下，I/Oと略記)に制御信号を送るLinkノード(Link Node, 以下LNと略記)が破壊，もしくは異常な制御信号を送るように変化した際には，バックアップ用に用意された幹細胞LNがその代替として動作するようになっている。また，LNを切り替える際に制御信号に間隙が入らぬよう，ある程度のbufferingも行う予定である。そのため，本システムは擬似的なreal time制御系と見ることも出来る。

このような自動復旧を果たす分散システムは，ごく普通に利用されるようになったEthernetとTCP/IPのみを用いて構築される。LN, I/Oにおいて実行されるエージェントはマルチスレッドプログラムとして実装され，大部分の通信をUDPを用いて非同期に行う。各Nodeは一般的なPCを用いるが，CPUはマルチスレッドに適したマルチコア系に移行しつつあり，これは将来において本システムのパフォーマンスの向上と大規模化に適したものといえる。

2. 全体構成

本システムの開発と，動作実験で使用したネットワークの物理的構成を図1に示す。一つのEthernet SwitchにLN, I/Oの役割と果たす各PCをつないだだけのごく一般的なものである。なお，ネットワーク構成は，UDPパケットが通過するものであれば何でもよい。ブロードキャストも使用しないため，Subnetに跨った構成でも理論上は問題ないはずである。

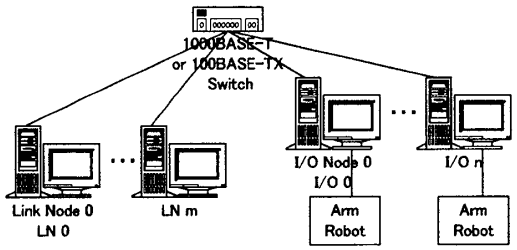

図1: 実験ネットワークの物理的構成

I/Oに繋がっている制御機器は何でも良いのだが，今回はRS-232C経由で繋がったアームロボットを使用する。

本システムでの前提は，全てのLN, I/Oが相互に繋がりうる物理的構成を持っていることである。その上において，正常に動作しているLNは完全グラフ構成で相互に結合され，常にお互いの動作を監視することになる。LN同士の完全グラフが構築された後，適切なアルゴリズムに従ってI/Oと接続するLNが選択され，制御信号が送信されるようになる。

I/O及びそれに接続したLNは制御系(Control System)を構成し，残ったLNは制御系に入ったLNのバックアップの役割を果たす幹細胞(Stem Cells)として動作する。

3. Link Agentの動作

LN上で動作するマルチスレッドプログラムを，Link Agentと呼ぶことにする。使用するスレッドは図2の通りである。

LN同士は完全グラフ構成に基づいてリンクを相互に張るため，LN数を $m \in \mathbb{N}$ とすれば，Send/RecvThreadは $m-1$ 生成される。I/Oと接続して制御系を構成するLNは，更にもう一つスレッド

*静岡理工科大学
*Shizuoka Institute of Science and Technology

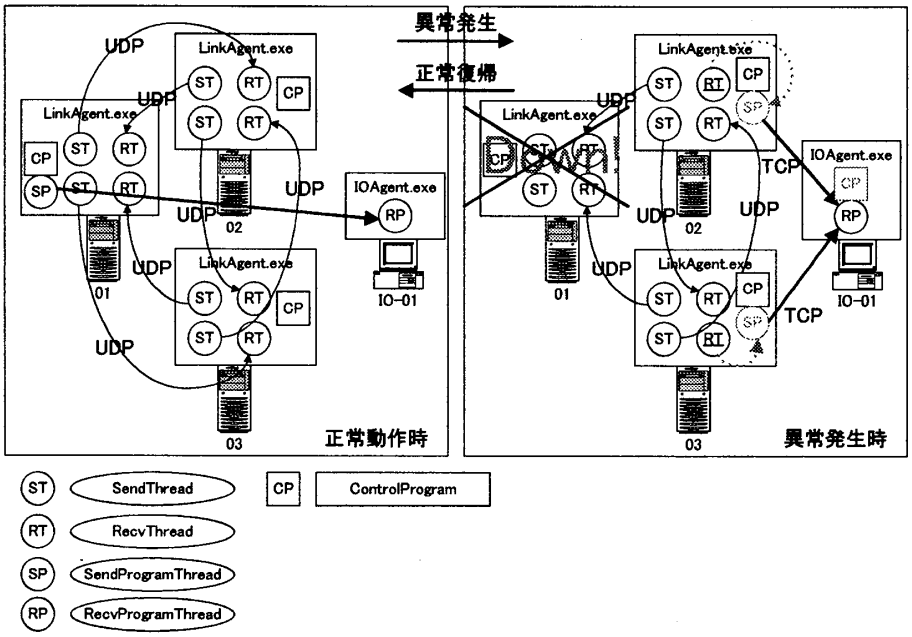

図 2: P2P 型制御システム及びスレッドの構成

が必要になるため，最大 m スレッドがエージェント内で非同期に動作することになる。

LN の動作は次のようになる。

1. 初期設定ファイルを読み込み，LN，I/O の IP アドレス，及び制御コマンドファイルを取り込み，幹細胞モードに移行する。
2. 自分以外の LN との通信を行うための Send/RecvThread を生成し，完全グラフリンクを構築。
3. I/O との通信を行う役割を担った場合は，I/O との通信を開始 (図 2 左)。
4. 幹細胞モードの LN は，他の LN とのリンクを維持し続ける。制御系の LN に異常が起こったらその LN の代替となるべく，I/O との通信を開始する (図 2 右)。

この一連の動作を行うエージェント群を，Windows XP Pro, Visual C++ 6.0 を用いて実装し，実際に以上動作時の LN の切り替わりがうまくいくかどうかの確認を行い，一応の成功を見た。詳細については講演時に示す。

4. 結論と今後の課題

今回の実験は，I/O が Fast Ethernet(100Mb/s に比べて著しく低速な RS-232C(9.6Kb/sec) に接続された 1 台の Arm Robot に対して行われたため，エージェント内に特別な通信用の FIFO バッファを置くことなく，目だったタイムラグを発せさせずに済んだと思われる。

今後の課題としては

- 複数の I/O に対応したバックアップシステム
- 異なる subnet 間での制御実験
- より高速な I/O 制御に対応できるよう，FIFO バッファ機構の実装

等が挙げられる。

謝辞

本研究は科学研究費萌芽研究 (課題 No.16656120) の助成を受けて行われた。関係者各位に感謝する。実装実験は本多隆一朗君 (2005 年 3 月卒業) の卒業研究の一環として行われたものである。

参考文献

[1] 大楫弘順, 自動修復分散制御システム, 特許出願書類 2003-135013.

[2] 藤本康孝・関口隆, 分散型プログラマブルコントローラ群による対故障制御, H13 年電気学会電子・情報・システム部門大会, TC5-3, 2001.

[3] L.Napper・江村豊 訳, Winsock 2.0 プログラミング, ソフトバンク, 1998.

[4] W.R.Stevens・篠田陽一 訳, UNIX ネットワークプログラミング 2nd ed., Vol.1, トッパン, 1999.