5G-03

分散ロボットシステムサーバのための応答性向上手法

中山　悟†　　中野　美由紀‡　　菅谷　みどり†
芝浦工業大学　理工学研究科　電気電子情報工学専攻†
芝浦工業大学　教育イノベーション推進センター‡

1.　はじめに

近年，安価なロボットの普及や無線への接続の普及により，複数のロボットを連携する分散ロボットシステムは、新たなサービス用途において大きな可能性を持つものとして改めて認識されている。複数台のロボットを利用したサービスにおいては，サービス提供者側は，個々のロボットを管理するために，その情報収集が必要となる。ロボットは、自律制御がある、ないに関わらず、移動や会話など、動的に変化する環境への変化が求められることから、その目的の達成についてデータを収集して確認するという要求は、システム構成上自然な要求である。

また、現在、我々の研究室では、見守りロボットシステムを開発している。これは、複数台の移動体ロボットが、それぞれがもつ役割に応じて対象者に支援を行うものである。各ロボットは、収集したデータを非同期でサーバに送信するが、役割によって必要となるデータが異なることから、サーバに送る頻度やデータの通信要求も様々である(表 1)。

複数台のロボットから情報収集する仕組みに関する議論は始まったばかりである。一般に従来のロボットは計算機とは異なり、ハードウェアやソフトウェアが固有であり、リソースやネットワークの信頼性が低いといった制約がある。また、サーバにおけるデータの処理について、スケジューリングによって，コアの最適配置を行い，計算コストの削減と応答性の向上を実現する取り組み[1]なども提案されている．しかし，ネットワーク自体の負荷が高い場合，そもそもサーバにデータが到達しない恐れがある．そのため、I0 サブシステムの制御を行うことが有効であると考えられる[2，3]。しかし、分散ロボットシステムに適用している例はあまりなく検証する必要がある。

本研究では，ネットワークが高負荷の場合でも，緊急の処理を要するデータの帯域幅を確保し，必ず受け付けて最短で応答することができる仕組みをロボット分散システムに適用し，その有効性を検証することを目的とする．実現のために，ネットワークレベルでの資源制御と，データグループの割り当ての仕組みにより構成する．

今回, 第一段階として, サーバクライアントモデルのプログラムを作成し、クライアントからサーバに送信する際に使用する各通信プロトコルで、データ転送にかかる時間の比較実験を行った．

表 1　見守りロボットシステムで収集するデータ

対象	ロボットが収集するデータ	サーバに送る頻度	通信要求
ロボット自身	動作状態	一定周期	
	電力	一定周期	
	位置	一定周期	
ロボットを介した人	生体	ストリーム（連続的）	
	位置	イベント時	高速、確実
	状態	ストリーム（連続的）	
他ロボット	位置	一定周期	

2.　提案

2.1.　概要

本研究では、ネットワークの負荷によらず、緊急のデータに対して迅速に応答することを目的とする。そこで、ネットワークレベルでの資源制御と，データグループの割り当てするミドルウェアとして、ダイナミックアダプテーションを行う仕組みを提案する。

本ミドルウェアを使用するにあたって、クライアントが送信するデータに対して優先度を付加する。帯域幅に対する優先度の低いデータの流量が、優先度の高いデータをサーバに接続されたすべてのクライアントが同時に送信するときの合計のデータサイズを上回る場合にリソース制限をかけることで、緊急のデータを常時受信可能な状態にする。受信したデータを、優先度に応じたキューに振り分けてスケジューラに処理させることで、緊急のデータに対する迅速な応答を可能とする。

本ミドルウェアは、クライアントからデータを受信して優先度ごとのキューに振り分けるアクセプタと、サーバとクライアント間で送受信しているデータの流量を監視しリソース制限をかけるモニタの 2 つから構成される。(図 1)それらの設計について以降で詳細に説明する。

2.2.　アクセプタ

クライアントからデータを受信するプロセスを異

†Shibaura Institute of Technology, Graduate School of Engineering and Science
‡Shibaura Institute of Technology, Center for Promotion of Educational Innovation

なる優先度グループとして作成する。それぞれのプロセス内で、使用する通信プロトコルごとに通信の受け口を作成し、クライアントから通信要求に応じた通信プロトコルでデータを受信する。

2.3. モニタ

cgroup を用いてアクセプタ内のそれぞれのプロセスを異なる優先度のグループとして管理する。優先度に応じてそれぞれのグループが使用可能な資源量を設定する。sar コマンドで出力したサーバが送受信しているデータ量が帯域幅に占める割合をみて、動的に各グループが使用できる資源量を管理する。

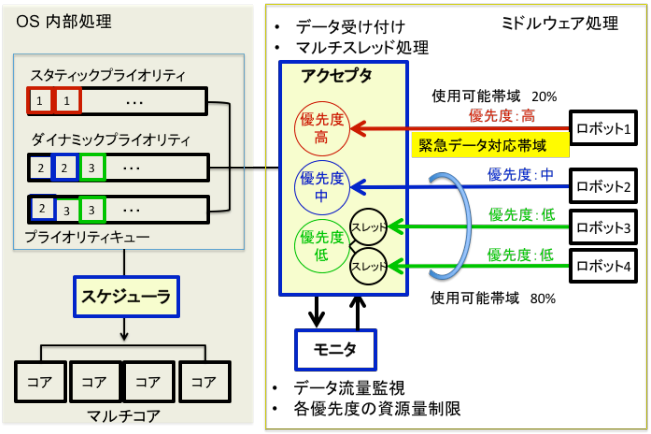

図 1　提案システム構成

3. 実験

OSI 参照モデルにおけるトランスポート層の通信プロトコルは複数存在する。そのうち主に使用される UDP と TCP について、データ転送時間の比較を行った。実験環境は表 2 の通りである。

表 2　実験に使用したマシン性能

	機種	OS	プロセッサ	プロセッサ速度	コアの総数	三次キャッシュ	メモリ
送信側	ThinkPad	ubuntu 14.04 LTS	Intel Celeron	2.00GHz	2	2MB	1.8GB
受信側	MacBook Pro	OS X	Intel Core i5	2.6GHz	2	3MB	8GB

UDP と TCP のそれぞれのプロトコルで、送信側と受信側の基本的なソケットプログラムを作成した。送信側がデータを送信し始めた時刻を T_{S1}、受信側からの返答を受信した時刻を T_{S2}、受信側がデータを受信した時刻を T_{D1}、返答を送信し始めた時刻を T_{D2} とする。送信側がデータを送信し、受信側から返答を受け取るまでの時間のうち、データの転送にかかった時間 $(T_{S2}-T_{S1})-(T_{D2}-T_{D1})$ を計測した。(図 2) 本実験以外は使わないローカルネットワークを使用し、ネットワークの負荷は一定であるとする。送信側が送信するデータサイズは 256 バイトである。

データ転送時間を UDP と TCP それぞれの場合で 12 回ずつ計測し、最悪値、最速値、平均、標準偏差を表 3 に示した。

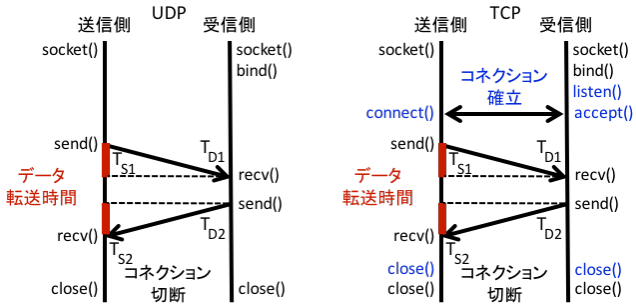

図 2　データ転送時間

表 3　各プロトコルでの計測結果

プロトコル	最悪値	最速値	平均	標準偏差
UDP	3510	630	2013	1002
TCP	4440	1430	2679	774

UDP では TCP よりもデータ転送時間が全体的に短くなった。最悪値と平均は約 20%、最速値は約 50%の速度差がみられた。しかし、標準偏差は UDP よりも TCP のほうが約 20%小さくなった。

UDP は単純なデータ送受信のみで、データ転送の信頼性を保証しないが、レイテンシの低減を重視したプロトコルである。一方で、TCP は信頼性を重視したプロトコルで、喪失したデータの再送や到着したデータの順序の並び替えなどの処理を行う。そのため、TCP は UDP よりもデータ転送時間が全体的に長くなったと考えられる。

4. まとめ

複数の種類のデータ送信を受け付けるサーバにおける、緊急のデータに対する応答性を向上させるための設計を示した。収集するデータごとの通信要求は様々であることから、基本的なデータ転送プログラムを実装し、評価を行った。UDP と TCP の各プロトコルで比較を行ったところ、最大 50%のデータ転送速度差がみられた。

今後、提案システムのミドルウェア部を実装し、優先度の異なる様々な形式のデータを収集したときの応答性の評価を行う。

参考文献

[1] Tao Zou, “Optimizing Response Time For Distributed Applications In Public Clouds”, A Dissertation Presented to the Faculty of the Graduate School of Cornell University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, January 2015

[2] Gaurav Banga, Peter Druschel and Jeffrey C. Mogul, “Resource Containers: A New Facility for Resource Management in Server Systems”, Proceedings of the 3rd Symposium on Operating Systems Design and Implementation New Orleans, Louisiana, February, 1999

[3] Eric W. Anderson and Joseph Pasquale, “The Performance of the Container Shipping I/O System”, SIGOPS ’95, December, 1995