

機器制御のための通信ミドルウェアの検討

増田 大樹 大谷 治之 落合 真一

三菱電機株式会社 情報技術総合研究所

1 はじめに

近年イーサネットが普及し安価となったことで、従来ネットワーク機能を持たなかった家電や工作機械など様々な機器においてイーサネット機能を持つものが増加しつつある。またその利用用途も拡大している。

本稿では上記のような近況を鑑み、工作機械やロボットの機体制御のようなリアルタイム処理に対してイーサネット経由で制御データを供給するシステムと、それを実現するための通信ミドルウェアの検討を行う。

2 対象とするシステム

今回我々は図 1のように、組込み機器とデータサーバをイーサネットで結合し、機体を制御する制御タスクがデータサーバに格納した制御用のデータを読み込みながら運転するシステムを検討している。制御タスクは 1.7ms ごとに動作するリアルタイムタスクであり、毎周期 256 バイトずつ制御データを読み込む。

従来のシステムでは組み込み機器内の主記憶等に制御データが格納され、制御タスクは API 呼び出しによって制御データを取得する構成となっていた。

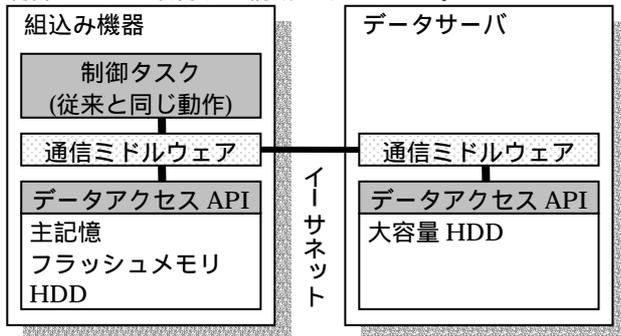


図 1 目的のシステム構成

図 1の構成を実現することで、データサーバの大容量 HDD を利用し、従来は容量の制約により組込み機器に入りきらなかった制御データを使用することができる。

本システムを実現するにあたり、組込み機器の構成は従来そのままとし、制御タスクは従来と同じ API 呼び出しによってデータサーバから制御データの取得を行う。

3 実装上の課題

上記システムを実現するためには、制御タスクの時間制約内に制御データを供給する API を実現しなければならない。そこで、制御データの一部をあらかじめデータサーバから読み込んでおく先読み領域を組込み機器の主記憶に設置する。API が先読み領域から制御データを取

得し制御タスクに渡すことで、制御タスクの時間制約を満たすようにする。

この方式を実現するには、API が実行された時点で必ず制御データが先読み領域に存在しなければならない。本章ではこの先読みしたデータ(以降：先読みデータ)を枯渇させないための課題について検討を行う。

3.1 先読みデータの消費

API が呼び出されるごとに先読みデータが消費され減少していく。そのため、先読みデータを追加する先読み処理がなければ先読みデータが枯渇してしまう。

先読み処理を別タスクで実現する方法も考えられるが、本システムでは組込み機器のタスク構成を従来そのままとするため、上記先読み処理を API の中で実現する。

API によって先読み処理を行うためには、次の処理をノンブロックで行わなければならない。

- ・ 追加要求の送信
組込み機器は先読み領域の空き具合に応じて、データサーバに制御データの追加要求を送る。(先読み要求)
- ・ 制御データの受信
組込み機器はデータサーバから制御データを受信し、先読み領域に格納する。

3.2 パケットの消失

通信路にイーサネットを用いた LAN を使うため、本システムはパケットの消失を考慮に入れなければならない。消失したパケットが届かない場合、必要な制御データが先読み領域に入らないため、先読みデータが枯渇することになる。従って、パケットの消失が発生した場合はそのパケットに含まれている制御データが必要となる前にパケットの消失を検出し再送を行わなければならない。また制御データの順序が保たれなければ機器が正常に動作することができないため、パケットの消失が発生しても、再送を行うのみでなく順序保証も行う通信をしなければならない。

このような信頼性を持った通信を行うために TCP の適用が考えられるが、TCP では分単位で再送が発生しない場合があり、その場合再送パケットを送信する前に先読みデータが枯渇する。そのため、本システムでは UDP を用いて短時間で再送を行う仕組みを構築する。

UDP を用いて再送を行うためには、次の事を実現しなければならない

- ・ UDP パケットの到達確認を行い、パケットの消失を検出する。
- ・ UDP パケットの順序を認識し、送信側と受信側の順序をそろえる

3.3 通信速度

本システムでは制御タスクによるデータ消費速度に間に合うように制御データを先読み領域に格納する通信速

度を実現しなければ、先読みデータが減少する傾向となり先読みデータが枯渇する。

さらに、パケットの消失が発生し再送を行った場合、消失が発生せずにパケットが到着した場合よりも通信が遅れる。そのためこのような異常が発生した場合は、パケットの到着までの間に先読みデータ量が通常よりも多く減少する。この減少した先読み量を回復する仕組みが無ければ、先読みデータの減少が蓄積し、やがて枯渇してしまう。

従って、本システムでは3.2節で述べた信頼性を持った通信の上で、パケットの消失が発生しても先読み量を回復できる通信速度を実現しなければならない。

4 システムの設計

4.1 先読み処理を行う API の設計

本 API は3.1節で述べた読み処理をノンブロックで実現するために、先読み要求の実行と、制御データの受信を非同期に行う。(図 2 上部参照)

[先読み要求の送信]

API が実行されると制御データが 1 周期分(256 バイト)消費される。そこで API 呼び出しの中で 1 周期分の先読み要求をデータサーバに送信する。この要求に対する回答は待たない。

[制御データの受信]

API が実行されたときに UDP の受信ソケットバッファに過去に送信した先読み要求に対する回答の制御データが届いている場合のみ受信処理を行う。このように先読み要求に対して即座の受信待ちを行わず、後に届いた時点で受信を行うことで、先読み処理をノンブロックで実行できる。API は先読み処理を行った後、先読み領域から制御データを取り出し制御タスクに渡す。

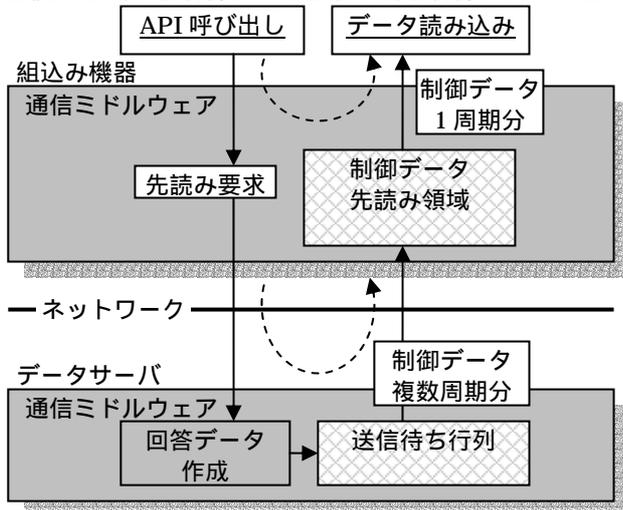


図2 ミドルウェア構成

4.2 通信処理の設計

本システムでは3.2節で述べたように UDP の上に信頼性を持った通信を構築する。順序保証と到達保証のために、通信パケットに管理番号を付けて管理を行う。

通信処理は次のように行う。(図 3)

- 1) 送信側はパケットに管理番号を付けて送信する。
- 2) 受信側は受信したパケットの管理番号を確認応答として返信する。

- 3) 送信側は送信から 10ms 以内に確認応答を受信できなかった場合、パケットが消失したとみなし再送を行う。

これにより順序保証と到達保証を行い、TCP と違って 10ms ごとに再送を繰り返すことができる。

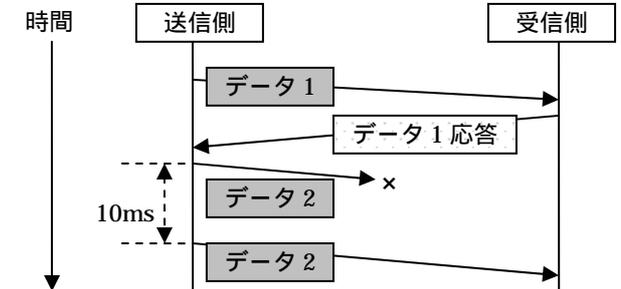


図3 通信方法

4.3 通信速度

本システムでは先読み領域を枯渇させないために、図 2 下部のような送信待ち行列を設け、制御タスク複数周期分の制御データを一度に送信する。これによりパケットの送信と確認応答を待っている間に先読み要求が溜まり、その間に減少した先読みデータ量を一度に回復することができる。

このように複数周期分のデータを一度に送れることで、例えば通信の往復が 3.4ms なら、送信から確認応答の受信までの間に 2 周期分の先読み要求が溜まり、2 周期分の制御データを一度に送ることができる。また、パケットロスが発生した時も再送待ちの間に先読み要求が溜まるのでパケットロス発生後早期に先読み量を回復することができる。

5 評価

本システムを次の環境を想定して、制御データの入ったパケットの消失を机上でシミュレーションする。

- パケットを送ってから応答が戻るまで 10ms
 - 制御データを最大 10 周期分(2.5kB)一度に送る。
- 計算した結果次の結果が得られた。
- 再送パケット到着までに経過する時間は再送まで 10ms と通信往復時間の 10ms を合わせて 20ms、この間に消費されるキャッシュは約 3kB。先読み要求により送信待ち行列には同じく 3kB 溜まる。
 - 3kB の先読み量減少を回復するためには、回復する間に消費される先読みデータも考慮すると 3 回の通信・30ms が必要となる。

従って本方式では 50ms に 1 回程度のパケット消失に耐えることができると見込める。

6 おわりに

本稿では工作機械の制御等、リアルタイム処理を行う機器に対し、ネットワーク経由でデータを供給する仕組みについて検討を行った。また、50ms あたりに 1 回程度のパケットロスに耐えられる見込みを得ることができた。今後実際の機器に組み込み評価を行っていく。

7 参考文献

- [1] J. Postel. Transmission Control Protocol. RFC 793, September 1981.