

CORBA による組込み機器監視システムの構築

大谷 治之[†] 増田 大樹[†] 落合 真一[†]

三菱電機（株）情報技術総合研究所[†]

インダストリアルオートメーションにおける組込み機器を監視端末とイーサネットで接続し、監視システムを効率良く構築する方法が求められている。インダストリアルオートメーションの監視システムでは、監視アプリケーションにおける表示データの即時性が求められる。表示データの即時性を提供するために、CORBA/IIOPによるone-way分散コールバック方式について提案する。また、ログを用いた表示データの回復機能についても提案する。本提案方式に従って実験用の監視システムを構築したところ、パケットロスの際も約100ミリ秒の即時性を持って監視データを受信し表示できることがわかった。

Building CORBA-based monitoring systems for industrial embedded controllers

HARUYUKI OHTANI[†], HIROKI MASUDA[†], and SHINICHI OCHIAI[†]

Information Technology R & D Center, Mitsubishi Electric Corporation[†]

Since it has been proved to be a cost-effective way to interconnect communicating entities, CORBA over Ethernet has recently got attention in industrial automation systems. In this paper, we describe our experimental remote monitoring systems for industrial embedded controllers based on CORBA over Ethernet. We implemented the one-way distributed callback and the function of log-based recovering for display data in order to meet the real-time and reliability requirements. The result of experiment shows that our implementation can be applicable to actual remote monitoring systems.

1. はじめに

近年、イーサネットの高速化および低価格化に伴ない、プロセスオートメーション、プラントオートメーション、ファクトリオートメーションなどのインダストリアルオートメーションにおいてイーサネットの利用が広がりつつある¹⁾。このような背景のもと、インダストリアル分野における組込み機器と監視端末をイーサネットで接続し、リモートから監視するシステムを効率良く開発することが求められている。本論文では、イーサネット上

でCORBA/IIOPを使った組込み機器の監視システムの構築方法について提案を行う。

2章ではインダストリアルオートメーションの組込み機器監視システムの要求について説明する。3章では、本要求を満たすため、one-wayによる分散コールバック方式の適用を提案する。単純にone-wayとするだけでは、要求を満たすことは難しく、それらを実現する追加の機能についても説明する。4章では、我々が構築した評価システムについて述べる。5章で提案について総括する。

2. 背景

2.1 動機

組込み機器群と監視端末をネットワーク接続する遠隔監視システムにおいて、経済性等の観点からイーサネットを採用する試みが行われている。このようなシステムを毎回ゼロから作成することは非常にコストがかかる。そこで最近では、単に伝送路にイーサネットを採用するだけでなく、上位通信層として分散オブジェクト技術 CORBA を用いることを視野に入れたシステム構築が行われつつある。

例えば、Amoretti らは CORBA を用いた組込み機器監視システム構築方法の評価を報告している²⁾。そこでは、ロボットを対象とした監視システムを、分散コールバック、Event Service、および Notification Service を用いて構築し、それぞれの利害得失の比較を行っている。但し、この評価は、高性能なパソコン上で CORBA とそのサービスを同時に動作させることで行っており、実際の監視システムの機器構成を忠実に模擬したものとは言い難い。

実際の監視システムは、振動、温度、電磁波ノイズ等の点で劣悪な環境で動作するのが通常であり、組込み機器はそれを考慮した設計がなされる。一方、監視端末は遠隔に設置されることから、通常のパソコンを適用可能である。従って、このような実際の機器構成を反映したシステム構築の評価が必要である。

そこで、本研究では組込み機器上で CORBA を直接動作させ、これらの動作状況をパソコン上の監視アプリケーションで監視するシステムを試作し、評価を行った。

2.2. システム構築上の課題概要

一般に、組込み機器の監視システムでは、表

示データの即時性が第一の評価基準となる。この表示データの即時性とは、表示されるデータが一定時間以上古くないことを意味する。

例えば、数値制御装置の軸の移動を監視するシステムでは、表示される軸の値は通常 100 ミリ秒以内に更新される必要がある。このようなシステムでは、実際の軸が移動するにつれて、表示する軸の値も同様に変化が必要がある。軸が移動しているにも関わらず、表示されるデータが 100 ミリ秒以上更新されない場合、オペレータは数値制御装置そのものに異常が発生したと判断する恐れがある。

システムバスと共有メモリを用いて監視端末と組込み機器を接続した、従来の監視システムでは、通信の高速性と信頼性がハードウェアによって保証されていた。このため監視端末から監視要求を出す単純なポーリング方式によっても、表示データの即時性を提供できた。

ところが、監視端末と組込み機器をイーサネットで接続し CORBA/IIOP を適用した場合、ポーリング方式によって表示データの即時性を提供することは困難である。

監視端末上の監視アプリケーションが即時性を満たすために短い間隔で要求を組込み機器に送信すると、1) 組込み機器上のサーバを飽和させ新しい要求を受け付けることができなくなる、2) ネットワーク帯域を飽和させる、3) 監視アプリケーションがポーリング以外の処理を十分に行うことができない、といった問題が発生する。

この問題に対する解決策として分散コールバック方式の適用がある³⁾。しかし、単純に分散コールバック方式を適用しただけでは表示データの即時性を提供することは難しい。

イーサネットは安価な反面ノイズに弱い。工場にはモータスタータなどの大きなノイズの

発生源があり、ノイズによるパケットロスが発生することがある。この場合、CORBA/IOP が用いる下位 TCP の機能により再送が行われる。そして、再送までに多くの時間がかかる場合、表示データの即時性を提供できない。

3. CORBA による組込み機器監視システムの設計

本章では、CORBA/IOP によって分散コールバック方式を用いた際、表示データの即時性を阻害する要因について述べる。次に、それを解決するため方式を提案する。

3.1 課題の詳細

表示データの即時性は TCP の再送と通信障害によって阻害される。また、コールバックに one-way を適用すると、障害からの回復が難しい。これらの課題について説明する。

(1) TCP の再送による即時性の阻害

信頼性を提供するためにコールバックオペレーションを要求・応答型の two-way として定義することが考えられる。two-way により、組込み機器は監視端末上のアプリケーションの処理結果を知ることができる。また、通信障害についてもコールバックオペレーションを呼び出す時点で検出することが可能である。更に、通信障害となったオペレーションについて、障害から復旧した後、再実行するなどの処置が行える。

しかし、コールバックオペレーションを two-way として提供すると、パケットロスが発生した場合、表示の即時性が提供できない。two-way において要求パケットのロスが発生すると、TCP の再送タイムアウトによって再

送が行われる。一般的に TCP の再送タイムアウトは百ミリ秒オーダーであり非常に長い。

また、監視端末が複数台ある場合は、two-way では別の問題を生じる。

組込み機器では設計に対する見通しを良くするため、作成するタスクは必要最低限であることが望まれる。このため、コールバックオペレーションを呼び出すタスクは通常 1 つである。1 つのタスクが複数の監視端末に対して two-way によるコールバックオペレーションを呼び出しを行い、ある監視端末との間で TCP の再送が発生すると、この間、他の監視端末に対するコールバックオペレーションの呼び出しができなくなる。これにより、正常に動作している監視端末において、監視データが長い間更新されなくなるという問題が発生する。

(2) 通信障害による即時性の阻害

断線などにより 1 つの監視端末との間で通信が行えなくなると、コールバックオペレーションの呼び出しは長い間ブロックする。この場合、TCP の再送が複数回行われ、TCP が再送をあきらめるまでの間、オペレーション呼び出しはブロックする。これにより正常に通信できている他の監視端末に対してコールバックオペレーションを呼び出せないという問題が生じる。

また、この問題は two-way だけでなく one-way においても起きる。通常、one-way のオペレーション呼び出しでは、ソケットバッファへのデータコピーと送信が完了すると即座に呼び出しがリターンする。しかし、断線などにより監視端末からの ACK が受信できないと、ソケットバッファ上に次々と送信データが溜まり、最終的にはオペレーション呼び出しがブロックする。

(3) 通信障害からの回復

コールバックオペレーションを one-way して定義すると通信障害からの回復が難しい、という新たな問題が生じる。

one-way では正確にどのオペレーション呼び出しを行った時点で障害が発生したかを特定することができない。なぜなら、ソケットバッファにデータを書き込む余裕がある場合、通信障害が発生していても、コールバックオペレーションの呼び出しは成功裏にリターンすることがあるからである。このため、障害から復旧する際も、どこから復旧していいかわからない、という問題が生じる。

3.2 課題解決へ向けての設計

(1) one-way コールバック

課題(1)を解決するために、one-way によるコールバックオペレーション定義を行い、コールバックオペレーションの周期的な呼び出しを行った。

コールバックオペレーションを one-way として定義することで、TCP の高速再送のメカニズムを活用し、高速に再送処理を起動することができる。

高速再送では、再送をタイムアウトによって再送を行うのではなく、古いシーケンス番号を受け取ることで、イベントドリブンに再送を行う。このためタイムアウトを待たずにパケットロスを検出した時点で即座に再送処理が起動することができる。

また、コールバックオペレーションの呼び出しを周期的に実行することで、高速再送が起動されるまでの時間を早めることが可能である。

(2) ノンブロッキングソケット

課題(2)を解決するために、適用している組み込み用の CORBA に対して、one-way オペレーションを実行する際、ソケットのノンブロッキング設定を行いブロックしそうな場合は即座に通信エラーを返すオプション機能を追加した。

一般的に、データが溜まることでソケットバッファへの書き込みがブロックすることをエラーとして扱うかどうかの判断については、使用するネットワークや監視端末のコールバック処理の性能に依存する。しかし、我々が想定する遠隔からの監視はイントラネットが対象であり、回線の速度や混雑および監視端末の処理能力について設計時に考慮できる。従って、正常時に使用するソケットバッファのサイズをチューニングすることが可能である。

(3) ログベースの復旧

課題(3)を解決するために、すべての監視データを書き込んでおくログを提供し、障害から復旧する際、監視対象としているデータをログから読み出し監視端末へと送る機能を提供した。

コールバックオペレーションを呼び出す監視タスクは、監視データを監視端末へ送信する前に、必ずログに書き込んでおく。

通信エラーが検出され接続が閉じられた後、監視タスクはコールバックオペレーション呼び出しの再試行を行う。この際、再び接続が確立し回復すると、ログを参照して必要とするすべての監視データを送信する。これにより、監視端末は最新の監視データを表示することができる。

4. 実装と評価

本提案の方式を実装した評価監視システムの構築を行った。以下に、CORBA のインタフェース定義を示す。

```
// リクエストした監視対象が存在しない
```

```
exception InvalidId {};
```

```
// コールバックインタフェース
```

```
interface CBHandler {
```

```
    one-way void push(in any data);
```

```
};
```

```
interface Monitor {
```

```
    // コールバックインタフェースと
```

```
    // 監視対象 ID および監視周期を登録
```

```
    void register_callback
```

```
        (in long id,
```

```
         in long tick,
```

```
         in CBHandler handler)
```

```
        raises(InvalidId);
```

```
};
```

register_callback オペレーションおよび push オペレーションの処理内容は次の通りである。

- register_callback オペレーションを用いて監視端末はコールバックインタフェース、監視対象の識別子(id)、監視周期(tick)を登録する。
- push オペレーションはコールバックオペレーションである。組込み機器上の監視タスクが指定された周期(tick)でこれを呼び出す。

図 1 に評価監視システムの構成を示す。組込み機器の OS には VxWorks5.4 を用い、監視端末の OS には Windows 2000 を用いた。監視端末と組込み機器は 100Base-T 全 2 重のイーサネットによって接続している。適用した

CORBA ORB は我々が組込み向けに開発したものである。

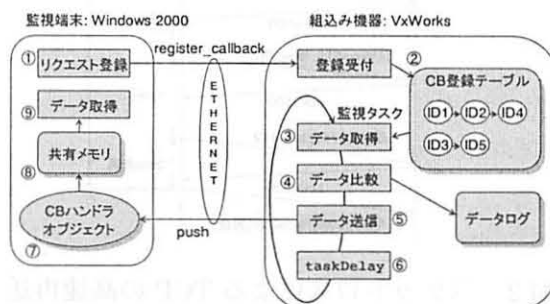


図 1 評価監視システムの構成

組込み機器上の監視タスクは 17 ミリ秒 (1tick)周期でコールバック登録テーブルを参照し、監視データを取得し、ログに書き込んだ後に、push オペレーションでデータを送信する(図 1 の③~⑤)。

(1) パケットロス時の即時性に関する評価

パケットロスが発生した際の表示データの即時性については、ポーリング方式と本提案方式の 2 種類の評価監視システムを構築し評価を行った。

我々の実験では、ポーリング方式において、要求がロスした場合は、TCP の再送が行われるまでに 200 ミリ秒以上を要した。Windows 2000 では SRTT(Smoothed Round Trip Time)に基づいて再送タイムアウト値を決定しているが⁴⁾、実際の再送タイムアウト値はこれよりも粗いことがわかった。

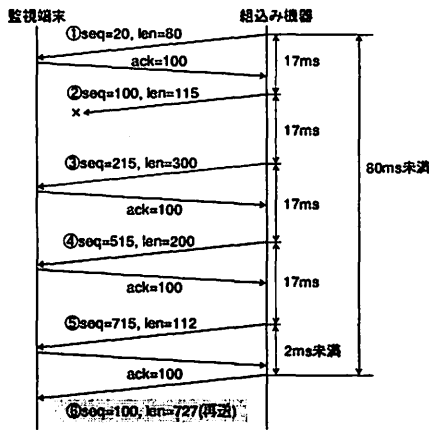


図2 パケットロスによるTCPの高速再送

一方、提案方式では監視データの packets がロスすると、packet が再送されるまでの時間は 80 ミリ秒以下に抑えることが可能であった。図2にTCPの再送の流れを示す。図2では②において組込み機器が送信した packet がロスしている。この packet のシーケンス番号は 100 である。次の周期③において、組込み機器は次のデータを送信するが、TCP レベルでは①のデータが到達していないことを示す ACK が組込み機器へ返される。④および⑤においても同様の ACK が返される。組込み機器は⑤においてシーケンス番号 100 以降のデータを直ちに再送する(⑥)。

このように監視端末上の監視アプリケーションは、通信オーバーヘッドを含めても 100 ミリ秒以内には、監視データを受信し表示データを更新できることがわかった。

(2) ログのオーバーヘッドに関する評価

監視データの合計サイズが 500 バイトの場合における、図1の③～⑤と④にかかる処理時間の比較を行った。

結果、③～⑤の合計処理時間は 1408 マイクロ秒であり、このうち④のログ書込みに 204 マイクロ秒かかることがわかった。これは全

処理時間の約 14% 程度のオーバーヘッドであり、割合としては小さい。

5. おわりに

組込み機器監視システムにおける表示データ即時性を満たすための CORBA による監視システムの構築方法について提案した。また、評価システムを構築し、パケットロスが発生しても約 100 ミリ秒以内に表データを更新できることを示した。

参考文献

- 1) D. Sterling and S. Wissler, The Industrial Ethernet Networking Guide
- 2) M. Amoretti, S. Bottazzi, M. Reggiani, and S. Caselli: Designing telerobotic systems as distributed CORBA-based applications, Proc. of the 2003 Intl. Symposium on Distributed Objects and Applications.
- 3) Douglass C. Schmidt and S. Vinoski: Distributed Callbacks and Decoupled Communication in CORBA, SIGS C++ Reports, Vol.8, No.10, October 1996.
- 4) D. MacDonald and W. Barkley: Microsoft Windows 2000 TCP/IP Implementation Details White Paper, 2000.