

第7回（社）計測自動制御学会

システムインテグレーション部門 講演会

7th SICE System Integration Division  
Annual Conference

SICE  
SI 2006

講演概要集

**SICE**<sup>®</sup>



2006. 12. 14-17 札幌コンベンションセンター

# RT コンポーネントの InPort/OutPort データ転送方法の多様化 –Raw TCP/IP Socket によるデータ転送–

安藤 慶昭 (産総研), 神徳 徹雄 (産総研), 安 芳次 (高工ネ研),  
 久保田 貴也 (産総研), 大川 猛 (産総研), 平野 聡 (産総研)

## Data transfer protocol diversification of RTC's InPort/ OutPort – Data transfer via Raw TCP/IP Socket –

\*†Noriaki ANDO, † Tetsuo KOTOKU, ‡ Yoshiji YASU,

† Takaya KUBOTA, † Takeshi OHKAWA, † Satoshi HIRANO

† National Institute of Advanced Industrial Science and Technology (AIST),

‡ High Energy Accelerator Research Organization (KEK)

**Abstract**— The RT-Component that we have proposed has data exchange port named InPort/OutPort. Currently InPort/OutPort are implemented on CORBA to transfer data. However CORBA, which has some overhead and is TCP/IP dependent protocol, does not always have to be used for data-centric communication. In this paper, we propose new InPort/OutPort which use Raw TCP socket to transfer data.

**Key Words:** RT(Robot Technology), middleware, network

### 1. はじめに

著者らは、RT のソフトウェアモジュール化を促進し、再利用性を向上させるためのプラットフォームとして RT ミドルウェアを提案してきた [1]。RT コンポーネ

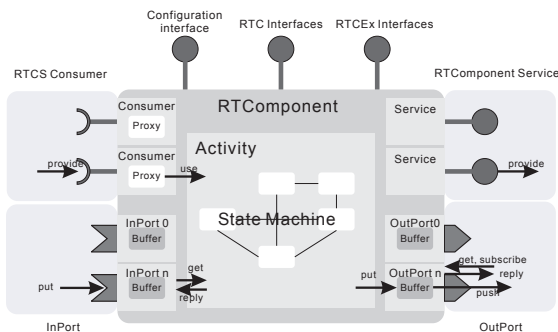


Fig.1 The architecture of RT-Component

ントは図 1 に示すように、コンポーネント間でデータの送受信を行う “InPort/OutPort” と、コンポーネント内部の詳細機能にアクセスするためのサービスポート “Service/Consumer” といった入出力ポートが存在する。本稿では、InPort/OutPort を多様な通信路や CORBA 以外のプロトコルで行う方法を提案する。

### 2. InPort/OutPort

著者らはこれまで、RT コンポーネントのオブジェクトモデルとして、図 1 に示モデルを提案してきた。

内部の主たるロジックであるアクティビティは、InPort に入ってきたデータを受け取り処理を行い、または処理したデータを OutPort から他のコンポーネントに送信する。

RT コンポーネントが、システム構成において比較的上位の処理をするコンポーネントの場合は、InPort/OutPort を通じて行うデータの交換は、時間的制約の厳しくない散発的なモノであることが多い。一方、

システムの比較的下位の処理をするコンポーネントの場合、InPort/OutPort は連続したデータの送受信のために使用され、時間的制約は厳しくなる傾向にある。

### 3. CORBA 通信の問題点

現在の RT ミドルウェアでは、全ての通信は分散オブジェクトミドルウェアである CORBA に依存している。しかしながら、CORBA はその複雑さゆえに、単純なデータのやり取りだけで比較すれば、Raw TCP Socket の通信よりもオーバーヘッドが少なからず存在する。

RT コンポーネントの InPort/OutPort は、接続に際していくつかの複雑なデータのやり取りを行うが、一旦接続が確立すると以降は単純なデータの送受信のみが行われる。従って、図 2 に示すように、データの送受信の部分を Raw TCP Socket で行えば、スループットやレイテンシの面でパフォーマンスが向上すると考えられる。

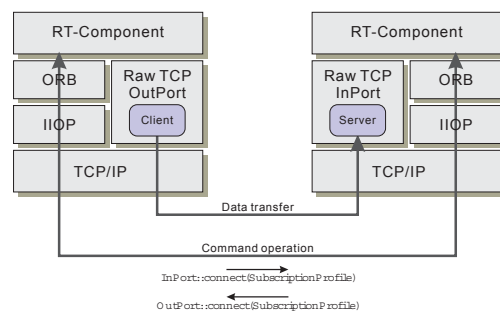


Fig.2 InPort and OutPort structure based on Raw TCP socket.

#### 3.1 多様な通信路への対応

CORBA は仕様上、特定のトランスポート・プロトコルに依存しない上位レベルの共通プロトコル GIOP (General Inter-ORB Protocol) が規定され、通常はこ

Table 1 Fast and Slow PC's specification.

	Fast PC	Slow PC
CPU	Pentium4 2GHz	Pentium II 200MHz
OS	Fedora CORA 4	Vine Linux 2.6
kernel	2.6.11	2.4.19
gcc	4.0.0	2.95.3

Table 2 The ratio of the raw socket's RTT to RTT of CORBA communication.

types	Slow PC			Fast PC		
	octet	long	double	octet	long	double
CORBA	282 $\mu$ s	284 $\mu$ s	278 $\mu$ s	99 $\mu$ s	100 $\mu$ s	99 $\mu$ s
1 item	0.56	0.55	0.55	0.85	0.83	0.83
4KB	0.70	0.89	0.88	0.98	0.97	0.99

れを TCP/IP にマッピングした IIOP(Internet Inter-ORB Protocol) が利用されている。原理的には、使用したい通信路のトランスポート層に対する GIOP マッピングを規定し実装すれば、アプリケーション側から見て透過的に他の通信路も使用することができる [2]。

しかしながら、新たなマッピング・実装の作成の手間や、事実上標準である IIOP を使用するメリットを考慮すると、データ通信部分のみ任意のプロトコルや通信路を使用する InPort/OutPort が実装できれば両者のメリットを得ることができると考えられる。Ethernet

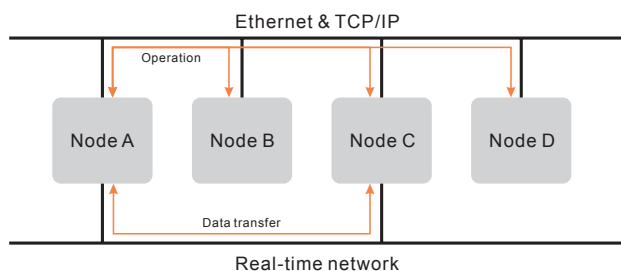


Fig.3 InPort and OutPort structure based on Raw TCP socket.

および TCP/IP を使用する限りにおいては、依然としてリアルタイム性は保証されないが、図 3 のように、仮にリアルタイム性を保証する通信路が利用可能な場合、こうした方法で InPort/OutPort を実装することで、互換性を損なうことなくデータ通信のパフォーマンスを向上させ、リアルタイム性を保証することが可能となるものと思われる。

### 3.2 CORBA パフォーマンス評価

CORBA, Raw TCP Socket それぞれによるデータ通信のパフォーマンスを比較するため、実験を行った。2 台の PC 間で、いくつかの種類・サイズのデータ (octet, long, double を 1 個および 4096byte 分) を伝送するプログラムを CORBA および Raw TCP を用いて実装し、RTT (Round Trip Time) の評価を行った。実験に使用したクライアント PC は、高速・低速の 2 種類を用意した。クライアント PC の仕様を表 1 に示す。CORBA の RTT を 1 とした場合の Raw TCP Socket の RTT の比を表 2 に示す。

表 2 から、“Slow PC” と “Fast PC” を比較すると、“Slow PC” においては Raw TCP Socket は CORBA に対して 55% から 89% 程度の RTT となっていることがわかる。“Slow PC” では “Fast PC” に比べ CORBA のマージングにかかる時間が支配的であるためと考

Table 3 Raw TCP InPort/OutPort Throughput.

	netperf	Raw TCP InPort/OutPort
4KB	507.88Mbps	509.27Mbps

えられる。また、データサイズが大きい場合は CORBA のオーバーヘッドよりもデータ伝送にかかる時間が支配的になっていることがわかる。

## 4. Raw TCP Socket InPort/OutPort

### 4.1 実装

InPort は Socket で直接データを受け取る TCP サーバとして、OutPort は InPort に対して Socket 経由で直接データを送信する TCP クライアントとして実装した。図 2 に示すように、InPort/OutPort は接続時に、サブスクリプションに関する情報 (SubscriptionProfile) を交換するが、ここにサーバのアドレスとポート番号を格納しクライアントに伝えることにより接続が開始される。

### 4.2 実験

実装した Raw TCP InPort/OutPort をそれぞれ持つ RT コンポーネントを作成し、2 つのコンポーネント間のスループットを計測した。各コンポーネントは GbE スイッチングハブで接続された 2 台の PC (Pentium4 3.2GHz, GbE NIC) 上で実行した。結果をネットワーク性能計測ツール “netperf” の計測結果と併記して表 3 に示す。

Raw TCP InPort/OutPort がほぼ netperf の計測値を達成していることがわかる。理論上のスループットは 1GBps であるが、netperf の比較からボトルネックは NIC もしくはドライバ等 OS 側にあり、Raw TCP InPort/OutPort にはオーバーヘッドが無いものと考えられる。

## 5. おわりに

本稿では、パフォーマンスやリアルタイム性および、通信路の多様性の観点から CORBA 以外のプロトコルを用いた InPort/OutPort について考察した。RT コンポーネントの InPort および OutPort のデータ伝送部分のみを、CORBA 以外のプロトコルで実装することにより、パフォーマンスの改善、通信路の多様化を実現する方法を提案した。

CORBA のオーバーヘッドを明らかにするために、CORBA および Raw TCP での通信のパフォーマンス測定・評価を行い、CPU 速度が遅い場合、およびデータサイズが小さい場合には CORBA のオーバーヘッドが問題となることが示された。一つの実装例として raw TCP/IP socket を用いた InPort/OutPort の例を示した。

今後は、RT コンポーネント内部のアクティビティやサブスクリプションのアーキテクチャを考慮したうえで、Raw TCP/IP socket を用いた InPort/OutPort のパフォーマンスを定量的に評価する予定である。

### 参考文献

- [1] 安藤, 末廣, 北垣, 神徳, 尹, 「RT 要素のモジュール化および RT コンポーネントの実装」, ロボティックスンポジウム予稿集, pp.288-293, 2004.
- [2] 石綿 陽一, 金広 文男, 比留川 博久, 齋藤 元, “Ethernet を用いた実時間通信方式の開発”, 第 22 回日本ロボット学会学術講演会, p.1C25, 2004