# 1Bb－4　レイヤ単位の並列処理方式を用いた通信プロトコルプログラムのHIPPIによる通信実験

佐藤 友実† 加藤 聰彦†† 鈴木 健二††
†電気通信大学 †† 国際電信電話(株)研究所

## 1. はじめに

伝送路の高速化に伴い通信プロトコルの高速処理方式が必要となっている。最近では、共有メモリ型マルチプロセッサ構成のワークステーションが広く普及しているため、このようなワークステーション上において、並列処理方式によりプロトコル処理を高性能化するための検討が重要であると考えられる。そこで筆者らは、1つのレイヤのプロトコル処理を個別のプロセッサに割り当てる、レイヤ単位の並列処理方式を検討している[1]。

筆者らは既に、市販の共有メモリ型マルチプロセッサ上に、本検討に基づいた評価プログラムを実装し、内部折り返しの評価を行なった結果、並列化のオーバヘッドは十分少ないことを確認した[2, 3]。次に、ネットワークを介して実際に通信を行わせた場合の評価を行うために、HIPPI(High Performance Parallel Interface)を使用して本方式の通信実験を行った。本稿では、その結果について述べる。

## 2. 並列処理方式の概要

筆者らのプロトコル並列処理においては、1つのレイヤのプロトコル処理を1つのスレッドで実現し、各スレッドを個別のプロセッサに割り当てている。また、各レイヤスレッドはキューを介して、共通にアクセス可能なバッファ領域上に作成されたサービスプリミティブをやりとりする(図1参照)。スレッドが互いに独立に動作してプロトコル処理をパイプライン的に実行可能なように、プリミティブの受渡しやバッファ管理において、スレッド間でなるべく同期を取らない方法を採用している[1]。

## 3. 実験構成と評価方法

### 3.1 実験構成

筆者らが提案した並列処理方式を、4つのCPUを有するSiliconGraphics社のONYXワークステーション(OS:IRIX 5.3 System V.4)上に実装した。実装に際しては、本方式に基づいた並列処理ライブラリを作成し、それを使用して評価用プロトコルプログラムを作成した。さらに、この評価用プログラムを、ワークステーションに実装されているHIPPIにインタフェースさせて、データの送受信を可能とした。実験においては、HIPPIのDestination(DST)ポートとSource(SRC)ポートに、1本のHIPPIケーブルの両端を接続し、折り返し転送を行った。

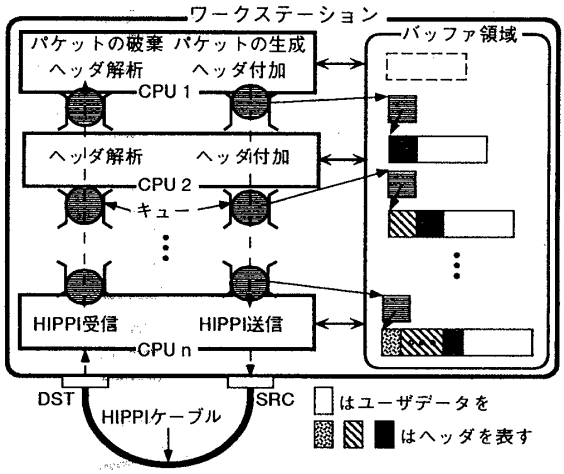


図 1: 評価用プログラムの構成

### 3.2 評価方法

性能評価は、図1に示す方法で行った。

(1) 並列処理の効果を明確化するために、同一のプロトコルを複数レイヤ積み重ねたプログラム構成を用いる。プロトコルとしては、トランスポートプロトコルクラス4(TP4)[4]を用いることとし、以下のような機能を実装することとした。

- 一本のコネクションにおけるデータ転送フェーズのみをサポートする。
- データ転送フェーズの機能の内、AK TPDUを用いたデータの受信確認、クレジットを用いたフロー制御、再送用のTPDUの保持およびチェックサムの設定と検査を実装する。なお、AK TPDUはクレジットの半分のDT TPDUを受信した時点で送信する。
- プロトコル処理のオーバヘッドの変化による影響を評価するために、ユーザデータのコピーを行わないバッファ方式[5]を用いた場合、memcpy関数によるユーザデータのコピーを行う場合、1バイトずつ参照してユーザデータのコピーを行う場合、およびユーザデータのコピーは行わず、チェックサム計算を行う場合の4つの処理方式を用いる。

(2) HIPPIの送受信のみを行うスレッドを設け、最下位に位置させることとした。また、最上位のレイヤにおい

---

"An experiment of communication protocol program using processor-per-layer parallel processing method through HIPPI interface"
Tomomi SATO†,Toshihiko KATO††and Kenji SUZUKI††
†The University of Electro-Communications
††KDD R&D Laboratories

ては、プロトコル処理の他に、送信するユーザデータ用のバッファの確保と、受信したユーザデータのバッファの解放を行う。

(3) 最上位レイヤのユーザデータサイズとレイヤ数を変えてスループットを計測する。また、参考のためにHIPPIのみの場合のスループットも計測する。なお、スループットは、受信側の最上位レイヤにおいて、ユーザデータの受信開始から終了までの時間で、転送ユーザデータサイズを割った値として計算した。

## 4. 結果と考察

図2に、データコピーを行わないTP4の処理を行った場合について、レイヤ数を1から3に変化させた場合のスループットの測定結果を示す。また、図3に、レイヤ数3の場合において、全レイヤがmemcpyによるデータコピー、1バイトずつのコピー、およびチェックサム計算を行なった場合のスループットを示す。なお、これらの図には、参考として、1つのスレッドでHIPPIの送受信処理のみを行った場合のスループットも示している。

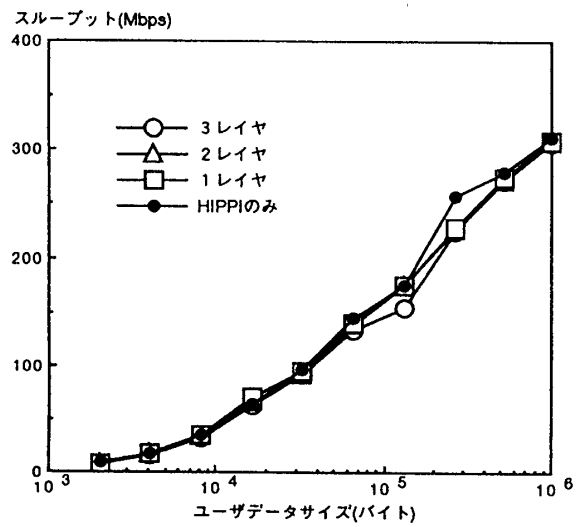


図 2: 評価結果 (コピーを行わない場合)

これらの結果より以下が考察される。

(1) 図2の結果は、レイヤ数が1、2、3の全ての場合において、ユーザデータサイズが64Kバイトのときスループットが約135Mbps、1Mバイトのとき約310Mbpsとなっている。これは送受信を同時に行った場合のスループットであるため、片方向の通信においてはその倍のスループットが得られると予想される。従って、筆者らの並列処理方式により、階層構造の通信プロトコルに対して、高速ネットワークを介した、高性能な通信を実現できると考えられる。また、この結果はHIPPIのみのスループットとほぼ一致しており、この実験においては、通信プロトコル処理のオーバヘッドがHIPPIの入出力処理に比べて小さかったと考えられる。

(2) 各レイヤの処理量を変化させた場合 (図3参照) については、同図には示していないが、レイヤ数を1、2とした場合についても同様の結果を得た。この結果から、本並列処理方式自身のオーバヘッドは無視できると考えられる。

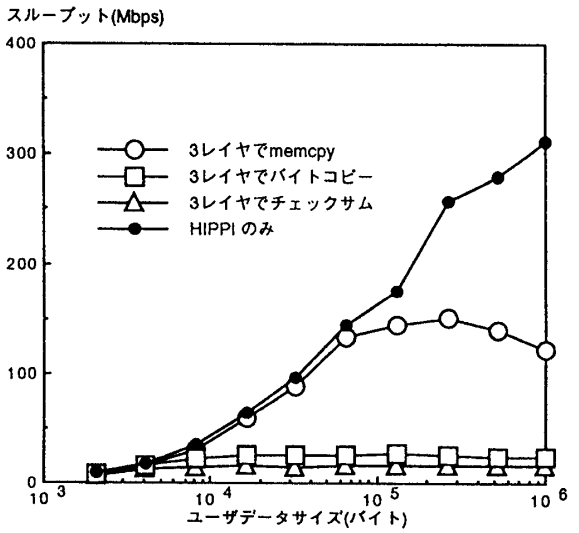


図 3: 評価結果 (各レイヤの処理負荷を変更した場合)

(3) 図3に示した各処理について、以下の評価を行うことができる。memcpyを行う場合は、ユーザデータサイズが60Kバイト以下のときは、HIPPIへの送受信の処理のオーバヘッドが大きいため、HIPPIのみのスループットと一致している。また、memcpyを含むプロトコル処理のスループットが約150Mbpsであるため、ユーザデータサイズが60Kバイトよりも大きい場合は、その値でほぼ一定となっている。一方、各レイヤで、バイトコピーとチェックサム計算を行った場合は、ユーザデータサイズが4Kバイトよりも大きい場合は、それぞれ、ほぼ25Mbpsと15Mbpsと一定になっている。これはソフトウェアによるバイトコピーもしくはチェックサム計算のオーバヘッドが大きいため、高速なプロトコル処理ができなかったことを示している。

## 5. おわりに

本稿では、レイヤ単位のプロトコル並列処理方式の性能を、HIPPIを用いた通信実験により評価した結果を述べた。本実験により、筆者らが提案する並列処理方式は、HIPPIを用いた通信においても、高速通信を実現可能であるということが明らかとなった。

## 参考文献

[1] 加藤、鈴木、"共有メモリ型マルチプロセッサを用いた通信プロトコルの並列処理方式," 情処マルチメディア通信と分散処理研究会, 69-17,March 1995.

[2] 佐藤, 加藤, 鈴木, "レイヤ単位の並列処理方式を用いた通信プロトコルプログラムの実装に関する検討," 情処第51回全国大会, 1E-3, Sept. 1995.

[3] 佐藤、加藤、鈴木、"レイヤ単位の並列処理方式を用いた通信プロトコルプログラムの実装," 情処マルチメディア通信と分散処理研究会, 73-19, Dec. 1995.

[4] CCITT X.224, "Transport Protocol Specification for Open Systems Interconnection for CCITT Applications." 1988.

[5] 加藤、井戸上、鈴木、"OSIプロトコル実装のためのユーザデータをコピーしないバッファ制御方式," 情処マルチメディア通信と分散処理研究会, 62-13,Sept. 1993.