マルチメディア通信と分散処理　46－11
（１９９０．　７．　１２）

OSI 7層ボードの設計

井戸上 彰　　加藤 聰彦　　鈴木 健二
国際電信電話株式会社 上福岡研究所

筆者らはパソコンやワークステーション用のOSIインタフェースとして、CPUを搭載するボード上でプロトコル処理を実行するOSI通信処理ボードの開発を行っている。このようなアプローチにより、本体計算機の負荷を軽減し高いスループットが得られるとともに、他機種への移植性も高めることができる。これまでにボード上でセション層までをサポートするOSI 5層ボードを開発したが、より高速なネットワーク上で高いスループットを実現するために、ボード上で様々なアプリケーション・プロトコルをサポートし、より高い処理能力を持つ汎用OSI 7層ボードの開発を開始した。本稿では汎用OSI 7層ボードのハードウェア設計およびソフトウェア設計について述べる。

Design of OSI 7 Layer Board

Akira Idoue　Toshihiko Kato　Kenji Suzuki
KDD Kamifukuoka R & D Labs.
2-1-15, Ohara, Kamifukuoka-shi, Saitama, 356 Japan

We have been developing OSI Communication Interface Board for personal computers and workstations. We think that our approach is an effective means to give these kind of computers the capability of the OSI communications at high performance, and to make it easier to develop and maintain the OSI products for various kind of personal computers and workstations. We have developed the OSI 5 Layer Board which supports OSI protocols up to the session layer. However, in order to achieve high performance on the higher speed network, it is required to support up to the application layer on the board. For this reason, we have started to develop the General Purpose OSI 7 Layer Board. This paper describes the hardware and software design of the OSI 7 Layer Board.

1. はじめに

開放型システム間相互接続(OSI)の標準化の進捗に伴い、大型計算機、ワークステーション、パソコンなどのさまざまな種類の計算機のための、OSI製品が広く開発されている。しかしこれらのOSI製品の多くは、OSIプロトコルの処理を計算機本体のCPUで実行されるソフトウェアとして実現しているため、次のような問題点が存在すると考えられる。

- MS-DOSパソコンでは、アドレス空間が640Kバイトに制限されているため、OSIの全レイヤのプロトコルを本体上のソフトウェアとして実装することが困難である。筆者らの経験ではX.25、トランスポート・レイヤのクラス0と2、セション・レイヤの全機能単位を実装するために、300Kバイトのプログラム領域とその他にバッファ領域を必要とする。従って640Kバイトの範囲でさらに上位レイヤを実装することは困難である。
- 計算機本体上で動作するソフトウェアとしてOSIプロトコルを実装すると、高いスループットが得られない。これはOSIプロトコル自身の機能の豊富さや複雑さ、短いパケットの受信による頻繁なプロセス切り替えのオーバヘッドなどに起因すると考えられる。
- ソフトウェアによるOSI製品では、ハードウェアやOSの異なった機種ごとに製品を開発する必要がある。OSIプログラムを異なる機種に移植することは、OSのシステムコール、内部的なデータ表現形式、コンパイラの機能などの違いにより困難であり[1]、また機種ごとに異なったプログラムを保守することも困難な仕事である。

そこで、筆者らはこれらの問題点を解決するために、CPUを搭載しOSIの通信処理を実行する「OSI通信処理ボード」の開発を行っている。これまでにその一つの形態として、パソコンやワークステーションを対象として、ボード上でセション層までをサポートする「OSI 5層ボード」を開発している[2]。このボードは上記の問題点を次のように解決することができる。

- MS-DOSパソコン上に26Kバイトのデバイスドライバを動作させるのみで、ボードの提供するセション層以下のすべての機能を用いることができる。
- OSI通信処理ボードでは、OSIプロトコル処理をボード上のCPUに分担させることにより、計算機本体の負荷を軽減する。また、受信確認パケットの処理やデータパケットのセグメンティング/リアセンブリングをボードで実行することにより、受信割込みによるプロセス切り替えを減らすことができる。その結果、開発したOSI 5層ボードでは、NECの16ビットCPU V50を用いて、64Kbpsの物理回線の90%以上の回線効率で、セション層以下の通信処理を行うことができる。
- OSI通信処理ボードを別機種に移植するためには
  ・バスインタフェース関連ハードウェアの変更
  ・ボード上のファームウェアのホストインタフェース・モジュールの変更
  ・本体計算機上のデバイスドライバの開発
  を行う必要がある。PC-9800用のOSI 5層ボードでは、ハードウェアのバスインタフェースに関連する部分は約10個の標準ロジックICのみから構成され、またファームウェアのホストインタフェース・モジュールは全体の1.5%程度である。従ってこれらの変更は非常に容易である。また、対象機種ごとに作成されるデバイスドライバも1K行程度の規模であり開発は困難ではない。

一方、高速ネットワークの導入が進むにつれ、さらに高いスループットのOSI通信を実現することが要求され、また本体計算機の通信処理の負荷をより軽減するために、応用層までのすべてのプロトコルをボード上でサポートすることが必要となっている。これに対処するため、筆者らはOSI 5層ボードのハードウェアを用いて特定のアプリケーション・プロトコルをサポートするOSI通信処理ボードの開発を行っている[3]。しかしより高いスループットで大規模なアプリケーション・プロトコルを実行するためには、高性能なCPUを搭載する新たなOSI通信処理ボードが必要である。そこで筆者らは、OSIの7層すべてを汎用的にサポートする「汎用OSI 7層ボード」の開発を開始している。本論文では、汎用OSI 7層ボードの設計について、開発の基本方針、ハードウェアの構成や使用するCPUなどのハードウェア設計、ボード上のカーネルの機能を中心としたソフトウェア設計について述べる。

2. 汎用OSI 7層ボードの開発の基本方針

汎用OSI 7層ボードを開発するために、次のような基本方針を立てた。

(1) 同一のハードウェア上で、FTAM、MHS等の複数のアプリケーション・プロトコルを同時に実行させる。

(2) PSPDN、PSTN、ISDNに加えて、LANや1次群ISDNなどより高速なネットワークも対象とするために、数Mbps程度までの伝送速度に対応可能とする。実際のボードのスループットは、アプリケーション・プロトコルまで含めて、数百Kbpsから1Mbps程度を目標とする。

(3) 下位層のハードウェアやソフトウェアは、対象とするネットワークに応じて開発するが、上位層のハードウェア・アーキテクチャやソフトウェアについては、ネットワークに依存せずに同一のものが使用できるようにモジュール化する(図1参照)。

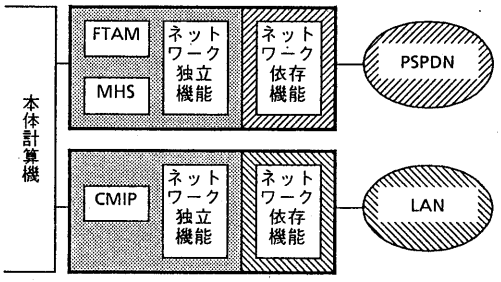

図1 汎用OSI 7層ボードの機能モジュール化

(4) 上位層のプロトコル処理を行うために処理能力の高いCPUを搭載する。

(5) 下位層のハードウェアやソフトウェアとしては、これまで開発したOSI 5層ボード、ISDNのBチャネル上でOSI通信をサポートするOSI対応ISDNボード[4]、CSMA/CDのLANのためのOSI対応LANボード[5]など開発済の資産を活用する。但し、当面はネットワーク・レイヤ以下としてX.25を対象とする。

3. 汎用OSI 7層ボードのハードウェア設計

3.1 ハードウェア構成

汎用OSI 7層ボードのハードウェア構成として、図2に示すような三つの候補をあげた。

(a) 高性能なCPUを1個用い、通信コントローラはCPUの配下にあるメモリに直接アクセスする方式

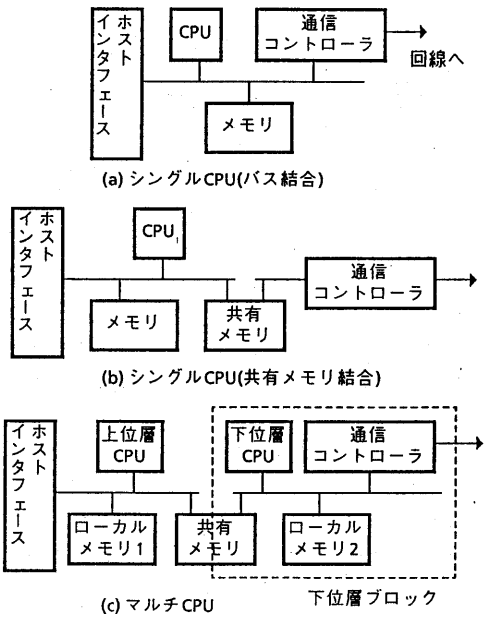

図2 汎用OSI 7層ボードのハードウェア構成案

(b) 高性能なCPUを1個用い、CPUと通信コントローラの間に共有メモリを実装する方式

(c) 上位層を実行する高性能なCPUと、通信コントローラとインタフェースして下位層を実行するCPUの二つのCPUを搭載し、両者を共有メモリで結合する方式

通信コントローラの処理能力への影響と、様々なネットワークに対する適応性について、以上の各構成を比較検討した。

(1) 通信コントローラの処理能力への影響について

10Mbps程度までのネットワークを対象とする場合には、CPUに比べて通信コントローラのクロック周波数は低く、メモリにアクセスするスピードが遅い。

(a) の構成では、通信コントローラがメモリにアクセスする際にバスを占有するため、CPUが停止する。CPUが停止する時間の割合は、ある一定量のデータを転送するのに通信コントローラが必要とするクロック数とクロック周波数によって定まる。例えばLAPBコントローラ、μPD72107の場合では、1バイトあたりの転送に2クロック必要である。μPD72107の動作クロック周波数は8MHzであるので、CPUが停止する時間の割合は1秒あたりに転送するデータ量に応じて表1に示すようになる。

表1 LAPBコントローラによるCPUの停止時間

転送速度 (bps)	必要クロック数	CPUの停止する割合(%)
64K	16K	0.2
1M	250K	3.1
4M	1M	12.5

表1から、4Mbpsの速度で連続的に送受信を行えば、25%の割合でCPUが停止することがわかる。CPUの停止時間は通信コントローラのメモリ・アクセスのスピードによって決まるため、特に高性能なCPUと、メモリ・アクセス・スピードの遅い通信コントローラを組み合せると、CPUの処理能力を無駄にしてしまう。

(b)の構成ではCPUと通信コントローラの間の共有メモリにより、通信コントローラに起因するCPUの停止をさけることができる。

(c)の構成では通信コントローラが下位層CPUに及ぼす影響については(a)と同様であるが、下位層CPUは、通信コントローラの制御や下位層プロトコルの実行など、比較的負荷の軽い処理のみを分担させ、上位層CPUで通信コントローラの影響を受けずにプロトコル処理に専念させることができる。この構成では共有メモリを通したデータ・コピーによるオーバヘッドが生ずるが、このオーバヘッドはメモリ・アクセス・スピードが高速のCPUを使用すればそれだけ小さくなるため、(a)の構成において通信コントローラによって処理能力が低下する割合よりは小さいと考えられる。

(2) 様々なネットワークに対する適応性

(a)の構成は実装が比較的容易であるが、CPUと通信コントローラのインタフェースが、通信コントローラの仕様に従うため、ハードウェア設計を様々なネットワークに対して共通化できる柔軟性は無いと考えられる。

(b)の構成でも(a)と同様に、CPUと通信コントローラのインタフェースを、通信コントローラごとに設計する必要があるため、様々なネットワークに対して共通に用いることができない。

(c)の構成では、下位層CPUには対象とするネットワークや通信コントローラに応じて最適なものを用いることができ、さらに上位層CPUと下位層CPUのインタフェースは、下位層によらず統一するすることができる。従って、上位層のハードウェア設計やソフトウェアの変更なしに様々なネットワークに適用できると考えられる。

以上の検討から、充分な処理能力を実現でき、様々なネットワークに適応する上でハードウェアの設計上柔軟性が高い(c)の構成を採用することとした。

3.2 CPUの選択

(1) 下位層CPU

先に開発したOSI 5層ボードやISDNボードのハードウェアや下位層のソフトウェア資産を有効利用するためには、これらが使用しているNEC V50(16ビット)を用いる必要がある。また、要求される処理負荷は比較的軽く、必要なメモリ容量もそれほど大きくないことから、16ビットCPUで充分であると考えられる。そこで、下位層CPUとしてV50を採用することとする。

(2) 上位層CPU

上位層のプログラム規模は下位層と比較してかなり大きくなると考えられるため、それに見合ったアドレス空間が必要である。また、複数のアプリケーションプロトコルをサポートすることやASN.1のエンコード/デコード処理などによって処理負荷も大きくなると予想され、高いスループットを得るためには高速な処理が要求される。このため、プログラム開発の容易さやコストも考慮して、一般的に使用されている高性能32ビットCPUを使用することとした。このようなCPUとして、モトローラ68030(68020)とインテル80386を選択の候補としたが、両者の間にはワードを表現する場合のメモリ上でのバイト順序が異なるという問題が存在する。これはプロトコル・ソフトウェアを開発する上でプロトコル要素の作成処理等に大きな影響を及ぼすが、これまでOSIソフトウェアを86系のバイト順序を前提として開発してきており、これまでのソフトウェア資産を生かすことができることなどから、80386を採用することとした。

80386は仮想記憶やマルチタスクをサポートするために、ページング、セグメント、タスク切り替え命令、特権命令、メモリ・プロテクション等高い機能を備えているが、ボード上ではこれらの機能のうち必要なだけを有効に利用する。どの機能を利用するかは、各プログラムモジュールの実現方法と密接な関係があるため4.において述べる。

3.3 上位層CPUと下位層CPUのインタフェース

上位層CPUと下位層CPUとのインタフェースの共有メモリに関して、そのサイズと、アクセス時の排他制御の大きさを決める必要がある。

共有メモリによるオーバヘッドは、メモリコピーのオーバヘッドと同程度であるため、共有メモリでは、送信と受信に対してそれぞれ1つずつのバッファを用意すれば充分であると考えられる。従って、共有メモリのサイズは、一回に転送されるデータの最大長を考慮して決定される。たとえばトランスポート層では現在最大TPDUサイズが8Kバイトであるため、下位層CPUでネットワーク層までをサポートする場合は送信用、受信用にそれぞれ約10Kバイト必要となる。一方、下位層CPUでトランスポート層までをサポートする場合はより大きなサイズが必要となる。

また排他制御の大きさについても、両方のCPUから同時に共有メモリにアクセスする確率は小さいと予想されるため、送受信用でそれぞれ独立した共有メモリを実装する程度とする。アクセス競合をより小さくするために、データの引渡し要求等を共有メモリとは別のハードウェア・レジスタや割り込み等によって通知することも検討する。

3.4 本体計算機とのインタフェース方式

ボードと本体計算機とのインタフェースに関しては、対象とするネットワークの伝送速度に対して充分高速な方式を採用する必要がある。先に開発したOSI 5層ボードでは、パソコン用では本体とボードとの共有メモリ方式を、ワークステーション用では本体メモリとボード・メモリの間のDMA転送方式を採用しているが、これらの方式によって数Mバイト/秒程度の高速なデータ転送が可能であるため、汎用OSI 7層ボードにおいても同じ方式を採用することとした。

4. 汎用OSI 7層ボードのソフトウェア設計

下位層CPU上のソフトウェアについては、開発済のOSI 5層ボードのソフトウェア資産を利用することとする。ここでは、上位層CPU上で動作するソフトウェアの設計について述べる。

4.1 ソフトウェアの実装方針

汎用OSI 7層ボードの上位層CPU上のソフトウェアに関しては、以下のような方針を立てた。

(1) プログラム・モジュールの柔軟なサポート

ボード上には、各レイヤやアプリケーション・レイヤの各サービス要素を実現する複数のプログラム・モジュールが実装される。ボード上のプログラム規模が大きくなるため、開発や保守はなるべく各モジュールごとに独立に行うことが望ましい。さらに、プログラム・モジュールを独立に開発させるため、モジュール単位のメモリ・プロテクションなどの、モジュールの保護機能を強化する必要がある。このため、ボード上に「OSI 7層ボード・カーネル」を搭載し、プログラム・モジュールを柔軟にサポートするようにする。

(2) 高い処理効率

高いスループットを得るために、ボード上での処理のオーバヘッドをできるだけ小さくし、プロトコル処理を効率よく実行する必要がある。このため、カーネルによるタスク切り替えやモジュール間の通信のオーバヘッドをできるだけおさえ、さらに各モジュールのきめの細かいスケジューリング機能を持たせる。また、各モジュールでもコピーを減らすなど処理のオーバヘッドをおさえ、モジュール間でやりとりされるプリミティブの構成も余分な処理を追加しないよう工夫する。

(3) ユーザ・プログラム・インタフェースの整備

ホスト計算機上のユーザ・プログラムが、容易にボードの機能を利用できるようにするためにホスト上にOSIボード・ライブラリを提供する。すなわちボード上のプロトコル・モジュールの機能を使用するための関数を用意し、ユーザにはボードとホストとのインタフェースや、やりとりされるデータのフォーマット等を意識せずにプログラミング可能とする。

以下に、ソフトウェア設計のキーとなるOSI 7層ボード・カーネルの機能について述べる。

4.2 OSI 7層ボード・カーネルの機能

OSI 7層ボード・カーネルは、各モジュールの実行制御(スケジューリング)、バッファ管理、モジュール間通信、タイマ管理等の処理を行い、これらの機能をシステム・コールとして各モジュールに提供する。ここではカーネルの設計にあたって、

各プログラム･モジュールの実行制御、スケジューリング、モジュール間通信について述べる。

(1) プログラム･モジュールの実行

各モジュールをカーネルからプログラムの実行単位として見た場合、次のような実行形態が考えられる。

① それぞれ独立したタスクとして実行する方式

② スレッドとして実行する方式

③ サブルーチンとして実行する方式

これらの方式はそれぞれ以下のような特徴を持っている(図3参照)。

① タスクとして実行する方式

この方式では次のような機能が用いられる。

- 各モジュールは独自のアドレス空間を持つ。
- 各モジュールのプログラム領域、データ領域およびスタック領域にセグメントが割当てられ、領域毎に不正なアクセスから保護される。
- カーネルがタスク間のデータのやりとりをサポートする。仮想記憶機能を用いて、やりとりされるデータを、各モジュールのアドレス空間にマップすることにより高速なモジュール間通信が実現可能である。
- 実行モジュールを切り替える際にコンテクスト･スイッチが必要である。
- 予め決まった数のモジュールのみが動作するため、モジュールの動的な生成/消滅は行わない。

これらの機能は、80386で提供されているタスク切り替え命令、ページングおよびセグメント等の機能を利用して実現することができる。

② スレッドとして実行する方式

全てのモジュールを同一のアドレス空間で実現し、モジュールの切替えのみをサポートする方式であり、次のような機能を使用する。

- 各モジュールは、プログラム領域およびデータ領域は共有するが、スタック領域はスレッドごとに用意される。
- 実行モジュールの切り替え時にコンテクスト･スイッチが必要である。カーネルの実装は、コンテクスト･スイッチの観点からは①と同等の処理が必要である。
- データ領域を共有しているため、スレッド間のデータのやりとりは同一アドレス空間内の共有データを使用することができる。

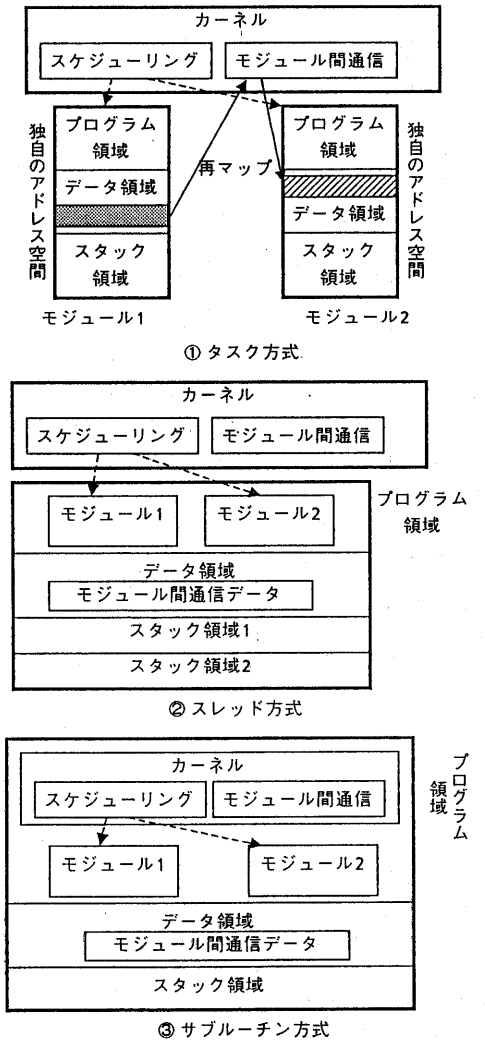

図3 モジュールの実現方式

- スレッド毎のメモリ･プロテクションは行わない。

③ サブルーチンとして実行する方式

各モジュールを順番にサブルーチンコールする方式で、現在のOSI 5層ボードで使用しているものである。

- プログラム領域、データ領域、およびスタック領域をすべて共有する。
- カーネルからは単にサブルーチン･コールを行うのみで、コンテクスト･スイッチは必要ない。
- モジュール毎のプロテクションは行わない。

プロテクション機能などによって各プログラム・モジュールを最も柔軟にサポートするという観点から、基本的にはタスクとして実行する方式を採用する。但し、特定のアプリケーション・プロトコルのみを実装する場合など、ソフトウェア規模が小さい場合には、スレッドまたはサブルーチンの方式でも充分であると考えられる。この場合、スレッドまたはサブルーチンのどちらを採用するかは、モジュール間でどの程度きめの細かいスケジューリングが必要となるかによって決まる。

(2) スケジューリング

カーネルはモジュールの実行順序を決定するスケジューラを持つ。スケジューリングの方法はモジュールの実行方法により異なる。

各モジュールをタスクまたはスレッドとして実行した場合においては、

① ある一定の時間間隔で強制的にスケジューラを起動し、コンテクスト・スイッチを行う方法

② カーネルのシステム・サービスが呼ばれたとき(プリミティブの入出力時等)のみスケジューラを起動し、コンテクスト・スイッチを行う方法

が考えられる。

一方、各モジュールがサブルーチンとして実行される場合は、以上のようなコンテクスト・スイッチはサポートされないため、

③ 各モジュールを常に一定の順序で呼ぶ方法

④ 各モジュールを呼ぶ順序を動作の状況に応じて変化させる方法

が考えられる。

コンテクスト・スイッチのオーバヘッドやスケジューリングの違いによってプロトコル処理の効率がどのように変化するかを調べるため、プロトコル処理とタスク・スイッチを模擬する簡単な評価プログラムを作成して実験を行った。評価プログラムは次のような構成とした(図4参照)。

- レイヤ・モジュールはデータリンク層からアプリケーション層までそれぞれ存在し、トランスポート層およびネットワーク層ではセグメンティング(送信側)またはリアセンブリング(受信側)の処理を模擬する。また、受信処理の場合はネットワーク層においてRRパケットを生成し、ウインドウのメカニズムをサポートする。
- レイヤ・モジュール間はキューを通して処理要求プリミティブを転送し、モジュール内ではプリミティブごとに指定された時間ウェイトする。
- 伝送路上の送信/受信は一定間隔の割込処理によって模擬する。割込ごとの送受信パケット長とパケット数を定め、送信の場合は送信要求キューから送信できる数だけのプリミティブを引取り、受信の場合はウインドウが閉じていない限り受信キューにプリミティブを生成する。

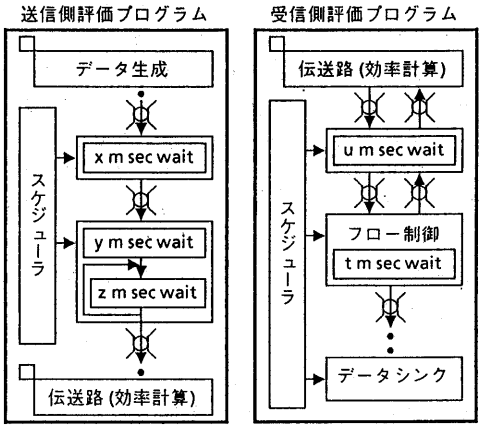

図4 スケジューリング評価プログラム

送信側の実験では、スレッド方式について、キュー出力時にコンテクスト・スイッチを行う方法と、キュー受信要求時に行う2つの方式を、サブルーチン方式について、すべての要求を処理する方式とキューから一つの要求を取り出して処理するごとにリターンする方式の効率を調べた。受信側の実験では、各モジュールはサブルーチン形式としたが、スケジューラが各モジュールを呼ぶタイミングを変更し、スループットの変化を調べた。

実験の結果から次のような傾向が得られた。

- 送信時においては、下位層のモジュールを優先的にスケジューリングした方が効率が良い。逆に過負荷時にアプリケーション層を優先的にスケジューリングするとスループットが下がる。
- キューへの出力時にコンテクスト・スイッチを行うというきめの細かいスケジューリングを行ってもあまり効果は見られない。
- 受信時には、ウィンドウを開くRRパケットがはやく送出されるようにスケジューリングするとよい。回線から連続的に受信している場合に、各レイヤがすべての要求を処理していると、RRパ

ケットの送出が遅れ、スループットが低下する場合がある。

この結果から、汎用OSI 7層ボードではあまり細かいコンテクスト・スイッチは必要ないが、ウィンドウを閉じないようにするなど、プロトコル処理の内容を考慮に入れたスケジューリングが必要となると考えられる。

(3) モジュール間通信の実現方法

先のOSI 5層ボードでは、隣接するレイヤ・モジュール間にそれぞれ送信用と受信用のキューを用意していたが、7層ボードにおいては、アプリケーション層では複数のモジュールが存在し、プレゼンテーションからの受信データはその内容によってそれぞれ別のモジュールに渡す必要があるなど、より複雑な構成となっている。そこで、それぞれのモジュールには専用の入力キュー(受信ポート)を用意し、キューへの送信要求はあて先モジュールと送出元モジュールの識別子を付加して、カーネル経由で行う方式とする。さらに、受信データを優先してスケジューリングしたり、優先データの送出を可能とするため、各モジュールには優先度の異なる受信ポートを2つ設けることとする。また、モジュール間通信のオーバヘッドをさけるため、通信のためのバッファは、送信側で割当てて受信側で解放することとし、キューにはバッファへのポインタをつなぐのみで、カーネル内ではデータのコピーは行わなものとする。このため、モジュールをタスクとして実現する場合には、カーネルはモジュール間通信のためのバッファ領域を割当て要求モジュールのアドレス空間にマッピングした後、受信モジュールがキューから取り出した段階でそのモジュールのアドレス空間にマッピングを切り換えるなどの処理が必要となる。

5. 考察

(1) ハードウェア構成の決定においては、作成済のOSI 5層ボードの活用が主な理由の1つとなった。その意味では、上位層CPUは、これまでに開発したボードに付加されて上位層のプロトコル処理を高速に行うプロトコルプロセッサと位置付けることができる。

(2) 本ボードは、10Mbps程度のまでの回線上で、最大1Mbpsのスループットを得ることを目標としている。このため、専用のプロセッサを使用せず、市販の32ビットCPUを用いてボードを作成することとした。OSI 5層ボードはV50(0.5MIPS程度)を用いて、セション層までの通信を100Kbps程度のスループットで実現可能であるため、4MIPSの80386とV50を用いることにより、アプリケーション層を含めて数百Kbpsのスループットを達成できると予想している。

(3) ボード上の各プログラム・モジュールは、通常の計算機上でのプログラムとは異なり、LAPBコントローラなどの通信制御デバイスに対する入出力中にCPUがアイドル状態になることはない。従って、各モジュールをサブルーチンとして実装するという単純なスケジューリングを用いても問題が生じないと考えられる。

6. おわりに

本稿では、CPUを搭載しOSIの7層プロトコルを高いスループットで実現する「汎用OSI 7層ボード」の設計について述べた。本ボードでは、先に作成したOSIボードの資産を活用でき、また低速な入出力デバイスにより高速なCPUが悪影響を受けないようにするために、下位層CPUと上位層CPUの2CPU構成を採用することとした。今後は、詳細なハードウェア設計や上位層CPU上のカーネルやプログラム・モジュール構成の検討を進め、NEC PC-9800シリーズならびにVMEバス用のボードの開発を行う予定である。最後に日頃御指導頂くKDD上福岡研究所小野所長、浦野次長に感謝する。

参考文献

[1]: 加藤他, "パソコンへのOSIトランスポート及びセションプロトコルの移植," 第32回情処全大, 6D-7, Mar. 1986.

[2]: 加藤他, "パソコン用OSI 5層ボードの開発," 信学技法, IN 89-22, June 1989.

[3]: 小花他, "パソコン用FTAMボードの開発," 1990年信学春季全大, B-654, Mar. 1990.

[4]: 石倉他, "OSI対応のパソコン用ISDNボードの開発," 1989年信学秋季全大, B-311, Sept. 1989.

[5]: 山崎他, "パソコン用LANインテリジェントボードの設計," 1989年信学秋季全大, B-243, Sept. 1989.