

通信ソフトウェアの標準化実装方式

4T-1

沖電気工業株式会社

篠塚 誠      新田哲二      松下 温

1. はじめに

近年、コンピュータネットワークの進展が著しく、システム間相互の通信が重視され、標準的なネットワークアーキテクチャとしてOSI標準化が進んでいる。これらの作業の進展に伴い、各種プロトコルの実装が始められており、今後、標準製品が急速に普及するものと思われる。

しかし、一方で既存のアーキテクチャによるシステムが一時になくなるとは考えられず、当分は既存アーキテクチャによる既存プロトコル製品とOSI標準製品が同一システム内に構築されると考えられる。

そこで、これらの製品製造において、統一した考えかたによって同一の通信プロセッサシステム上に実装する方法を考察したので報告する。

2. 通信ソフトウェアのモデル

今回対象とする通信ソフトウェアは、通信プロセッサの通信仮想オペレーティングシステム上にプロセスとして実装される。したがって、プロセスは実オペレーティングシステムの環境に依存しない。

プロセスはレイヤとは異なっている。すなわち、一つのプロセスが複数のレイヤにまたがったり、上位レイヤ、下位レイヤの機能を部分的に取りこんでいたりする。

そこで、まず、レイヤと区別をするために、通信プロセス階層という考えかたを導入する。レイヤとの対応はプロセス毎に考察して明らかにする。

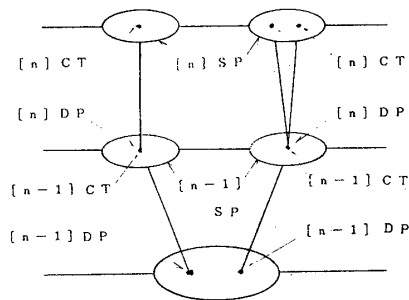
OSI 階層

プロセス階層

|                    |                       |
|--------------------|-----------------------|
| Application layer  | Application-processes |
| Presentation layer | Online-processes      |
| Session layer      | S-process             |
| Transport layer    | D-processes           |
| Network layer      | E-process             |
| Datalink layer     | Datalink process      |
| Physical layer     | ccc firmware          |

次に、プロセスをモデル化する。[n] プロセス内の要素としては、実装を考慮しながら、コネクション端点、SAPなどの概念に相当するものを定義した。また、プロトコルによってはコネクションの定義がないが、その場合は、サービスインターフェース上でコネクション確立/解放をしているようにみせかける。プロセスがレイヤと一致するとき、プロセスのモデルの要素は以下のような対応関係になる。

- [n] CT (Connection Terminal)  
(n) コネクション端点に相当する
- [n] SP (Service Port)  
(n) SAPに相当する。
- [n] DP (connection Drive Port)  
(n-1) コネクション端点に相当する



3. 標準サービスプリミティブ

既存プロセスには多くの種類があって、これらのすべてのサービスを標準化することは現実的に難しい。

そこで、次の観点からサービスの統一化の範囲を限定した。

- (1) コネクション確立/解放に関するサービスは統一する。
- (2) 実際にプロトコル構成の変更を行うプロセス同士のサービスについては、パラメータに関しても統一化を進める。
- (3) 既存プロセスを整理した結果セッションレイヤに相当すると考えられるものが多いので、セッションサ

一サービスプリミティブを参考にしてサービスの統一化を図る。

#### 4. プロセス管理

システムがうまく動作するためには、システムの開始/終了動作や資源の監視、異常状態の処置などの管理が必要である。また、ネットワーク全体をうまく動作させるためには、個々のシステムを遠隔地から管理することも必要になる。

これらの管理を行うためには、まず、システム内のプロセスが階層毎に制御できなければならない。しかし、現状のプロセスは、階層毎の制御機能をもっておらず、最上位層に対してサービスが発行されると各階層が連動するように設計されている。コネクション確立/解放や障害状況の報告や障害回復などの制御はシステム生成の条件で決まる。こうした方法は実装が容易ではあるが、柔軟性に欠けていた。

そこで、高位プロセスが低位プロセスを管理する従来からの形態を基本とし、さらに、障害管理の一部および保守機能の一部について各プロセス階層にわたるマネージャプロセスが介入する実装方法とした。ネットワークの資源管理については通信プロセス以外のソフトウェアで実現する。そのために高位プロセスを通じて低位プロセスのサービスが利用できる方式とする

#### 5. プロセス階層の動作

プロセスを独立させ、それぞれの階層のプロセスのサービスを利用できるように、プロセス間のコネクション毎のサービスの利用者 (DP) とサービス提供者 (CT) の対応づけはコネクション確立/解放を伴う標準サービスで行う。

おもな標準サービスによる動作の概要は次のとおりである。

##### (1) SPオープン

APが通信を開始するとき、[n] SPオープンサービスを発行する。これにより [n-1] 以下のプロセスのSPがオープンされ [n-1] プロセス以下のコネクションが同時に連動して確立される。サービスが正常に終了すると [n] SPからの発呼および [n] SPへの着呼が許可されて [n] コネクションの確立/解放を行うことができる。

プロセス階層 [n] を指定することができ、これにより階層毎のコネクション管理ができる。

##### (2) SPクローズ

APが通信を終了するとき、[n] SPクローズサービスを発行して [n] SPを通じた発呼、および着呼を禁止し、同時に [n-1] 以下のコネクションを解放する。このとき、コネクションが多重化されていれば、それ以下のコネクションは解放されない。

##### (3) コネクション確立

オープンしたSPを通して [n] 以下のコネクションを連動して一度に確立する。既に確立されているコネクションがあれば多重化する。

##### (4) コネクション解放

[n] 以下のコネクションを連動して解放する。[n-1] 以下のコネクションで多重化されているものがあれば、それ以下のコネクションは解放されない。

##### (5) エラー

プロセスが異常を検出するとコネクションの利用者およびマネージャプロセスにエラーを通知する。マネージャプロセスに対する通知はハンドリングされてネットワーク資源管理プロセスがあればそこへ通知される。

これらの動作は、前述のCT, SP, DPによるモデルで表すことができる。さらに、モデルの要素をテーブルに置換えれば、それぞれのテーブルを中心とした処理を考えることができる。

#### 6. おわりに

本実装方式によって、プロセス階層の積み換えや入換えが可能になり、プロセスをプロトコル構成毎に造る必要がなくなり、プロトコルの種類だけプロセスを用意するだけで柔軟なプロトコル構成による開放型システムを構築できるようになった。また、プロトコル毎に考えられていたネットワーク資源およびアクセスに対する考えかたを統一化することができ、また、通信ソフトウェアの利用者に対して統一的なアクセス方法を提供できるようになった。

#### 7. 参考文献

- 1) OKITAC system50V 標準TCP作成手引書
- 2) PX Csystem 標準TCP論理インタフェース仕様書
- 3) 勅使河原: 開放型システム間相互接続 (OSI) の参照モデル、情報処理, Vol. 26, No. 4, pp. 299-309 (1985)