Vol. 30 No. 4 情報処理学会論文誌 Apr. 1989

ショートノート

異種プロトコル共存時のプロトコル処理モジュール構成法†

竹内宏典†† 工藤明彦††

プロトコル TCP/IP と X.25 を例として，UNIX ネットワークシステムにおける異種プロトコルの実装手法について述べる．本手法により，プロトコル処理モジュール間の独立性を確保しつつ，プロトコルの整合が可能となり，性能は実用に耐えうることを示す．また，ネットワークアプリケーションプログラムの通信路からの独立性を確保できることを示す．さらに，この実装手法を一般化して，プロトコルファイバ，プロトコルスイッチという概念を用いたプロトコル処理モジュール構成法を提案する．

1. はじめに

UNIX* 分散処理システムの多くは，プロトコル TCP/IP とローカルエリアネットワーク (LAN) により構築されてきたが，今後，それらのシステムを DDX 網等の広域ネットワークを介して接続した広域分散処理システムへと移行することが予想される．

筆者らは，LAN および広域網の両方に適用可能な UNIX ネットワークシステム構成法の検討を行ってきた[1),2)]．このようなシステムでは，異種プロトコルをうまく共存させ，アプリケーションプログラムをこれまでと同様に動作させる必要がある．

本論文では，DDX (PS) 網を介した TCP/IP の実現方法を例として，TCP/IP と X.25 の整合にあたっての問題点を明らかにし，それを解決するための手法について述べる．さらに，その手法を発展させた異種プロトコル共存時のプロトコル処理モジュールの構成方法を提案する．

2. 前提条件

研究の前提条件を明確にしておく．

(1) TCP/IP および X.25 を実現するプロトコル処理モジュールは，すでに製品化され流通しているソフトウェアを利用する．かつ，TCP/IP は，socket インタフェースを備えている．

(2) LAN (イーサネット) を介した TCP/IP による UNIX 分散処理システムは，このうえで走行するネットワークアプリケーションプログラムも含めて，すでに稼働している．

3. X.25 と TCP/IP の整合

本章では，X.25 と TCP/IP の両プロトコルを整合させるうえでの具体的な問題点を明らかにし，筆者らがとった解決策について述べる．

3.1 インタフェースプロトコルの導入

プロトコル TCP/IP では，相手局と自局の間にセションを開設するのは TCP であり，IP はデータグラム転送を行うだけである．一方，プロトコル X.25 はバーチャルコール (VC) を用いるため，相手局との通信に先立って呼を設定する必要があり，データ送受信のほかに呼設定，呼解放などの呼制御が必要である．X.25 の呼制御を行う方法として，①アプリケーションプログラムに呼制御のインタフェースを提供し，アプリケーションプログラムで行う方法，②TCP のコネクション制御セグメントを X.25 の呼制御にマッピングして行う方法，③ IP に X.25 の呼制御を行う機能を持たせる方法，④ TCP/IP と X.25 の間に X.25 の呼制御を行うレイヤを設け制御する方法，が考えられる．このうち，①の方法はアプリケーションプログラムに DDX (PS) 網を意識させることになり，③の方法は IP を実現するプロトコル処理モジュールの改造を伴いかつ IP に通信路の意識を持たせることとなり，設計条件に反する．また，②の方法は IP 以下のレイヤで TCP セグメントの内容を参照することになり，プロトコルの階層構造上問題がある．そこで，X.25 の呼制御を IP より上位に隠ぺいすることが可能であり，筆者らの設計方針にも合致する④の方法に基づき，TCP/IP と X.25 との整合を図るイン

† The Structure Design of Protocol Modules by HIRONORI TAKEUCHI and AKIHIKO KUDOH (NTT Communications and Information Processing Laboratories).

†† NTT 情報通信処理研究所

* UNIX は，AT&T で開発された OS である．

タフェースプロトコルを導入した.

3.2 インタフェースプロトコルの機能

(1) 呼制御

呼制御として,(a)呼設定機能,(b)着呼機能,および(c)呼解放機能を規定した.(a)は,IP からのデータ送信要求が発生した時に呼設定済みか否かを調べ,呼設定済みであればデータ送信を行い,呼が未設定であれば呼設定後データ送信を行う機能である.(b)は,相手局からの着呼要求に対し着呼確認応答を返して呼設定を行い,以降受信データを IP に転送する機能である.(c)は,(ア)全論理チャネル使用中に(a)により新たな呼設定を行うことが必要になった場合,最も長くデータ送受信が行われなかった呼を解放し,(イ)相手局からの呼解放要求に対し呼解放応答を返し,(ウ) UNIX システム立ち下げ時に呼解放を行う機能である.

(2) アドレス変換

プロトコル TCP/IP は,相手局のインタネットアドレスを指定して IP パケットを送出する.一方,プロトコル X.25 は,相手局との間に設定されたコネクションの論理チャネルグループ番号,および論理チャネル番号を必要とする.論理チャネルグループ番号と論理チャネル番号は,相手局との間に呼を設定する際に呼に一意に付与されるが,呼の設定に当たっては相手局を識別するための相手局 DTE 番号(相手局の DDX 加入者番号)が必要である.このため,TCP/IP と X.25 を整合させるためには,インタネットアドレスと DTE 番号,および論理チャネルグループ番号,論理チャネル番号の変換が必要である.これを解決するために,X.25 と TCP/IP とのアドレス変換を規定した.

筆者らが実装したアドレス変換機能は,UNIX ファイルとして作成したデータベースを参照することにより行うこととし,アドレスの追加・削除が容易に行える.さらに,IP ルーチングに関しても,アドレス変換機能により一部実現した.

3.3 インタフェースプロトコルの実装と評価

2章で述べたように,研究対象のシステムでは,X.25 および TCP/IP は,各々独立したプロトコル処理モジュールとして実現されており,インタフェースプロトコルについても,別のプロトコル処理モジュールとして実現した.ソフトウェア構成を**図1**に示す.

以上述べた手法により TCP/IP と X.25 を整合したシステムを①本手法によるオーバヘッド,②ネットワークアプリケーションプログラムの通信路からの独立性の確保,③プロトコル TCP/IP と X.25 の独立性の確保,という観点から評価した.①については,LAN および DDX 網を介したファイル転送の実効速度を比較し評価した.その結果を**表1**に示す.②については,**表2**に示すネットワークアプリケーションプログラムが通信路に関係なく動作することを確認した.③については,TCP/IP および X.25 プロトコル処理モジュールのプロトコル整合に関する修正量がゼロであることを確認した.

評価結果から以下のことが主張できる.

表1 ファイル転送速度
Table 1 File transfer rate.

通信路	伝送速度	ファイル転送速度
LAN	10 Mbps	28.9 kB/秒
DDX (PS)	48 kbps	1.8 kB/秒

表2 ネットワークアプリケーションプログラム
Table 2 Network utilities and commands.

プログラム名	機能
ftp	DARPA File Transfer Protocol
telnet	DARPA Virtual Terminal Connection
rcp	Remote File Copy
rlogin	Remote Login
rsh	Remote Shell
rwho	Who is Logged In
ruptime	Host Status of Local Machines

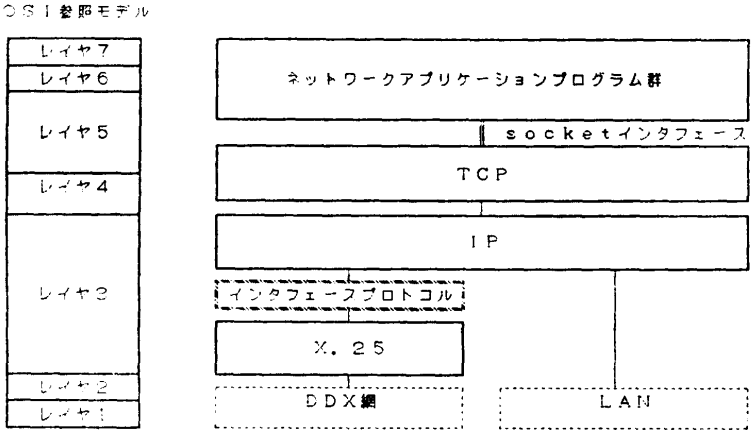

図1 TCP/IP と X.25 が共存する場合のソフトウェア構成
Fig. 1 Software structure.

Ⅰ. 本手法により，プロトコル TCP/IP と X.25 の整合が可能となり，性能は実用に耐えうる．

Ⅱ. 本手法により，ネットワークアプリケーションプログラムは，LAN と DDX 網を意識することなく，両通信路上で同等に機能する．

Ⅲ. 本手法により，各プロトコル処理モジュールの独立性を確保できる．

4. プロトコル処理モジュール構成法

これまで，TCP/IP と X.25 を例として異種プロトコルの実装手法について述べてきたが，より高度な分散処理システムを構築するためには，標準化の進む OSI プロトコルや ISDN 等の新しいメディアに対応するためのプロトコル実装手法の確立が重要である．そこで本章では，3章で述べた実装手法を発展させ一般化した異種プロトコル共存を考慮したプロトコル処理モジュール構成法を提案する．

プロトコル処理モジュール構成法の一般化のためには，

(1) プロトコル処理のモジュラリティを確保するためのプロトコル処理のモジュール間インタフェースの規定，

(2) プロトコルやデバイスドライバの組合せの表現方法，

を明確にする必要がある．筆者らは，後述するプロトコルファイバおよびプロトコルスイッチという二つの概念の導入により，これらの問題を解決した(図2)．

4.1 プロトコルファイバ

プロトコルおよびデバイスドライバは階層構成となっており，この階層をファイバレイヤと呼ぶ．プロトコルおよびデバイスドライバの組合せを考えた時，ファイバレイヤ1，2，…に対して各々ひとつのプロトコルあるいはデバイスドライバが決まる．そこで，ファイバレイヤの順に，プロトコルファイバテーブルにプロトコル ID (プロトコルおよびデバイスドライバを特定するための識別子) を登録しておき，ファイバレイヤ番号とファイバ ID (プロトコルおよびデバイスドライバの組合せの識別子) より，次に呼び出すべきプロトコル処理のプロトコル ID を得る．

4.2 プロトコルスイッチ

上位あるいは下位プロトコル処理関数の呼び出し，デバイスドライバの呼び出しは，関数名を直接コーディングするのではなく，プロトコル ID を基にプロトコルスイッチテーブルに登録されている関数を求め，それを呼び出す．これは，4.2 BSD 等で用いられているプロトコルスイッチを拡張したものである．

4.3 プロトコル処理モジュール構成法

プロトコルファイバおよびプロトコルスイッチを用いたプロトコル処理モジュール構成法の概念を図3に示す．

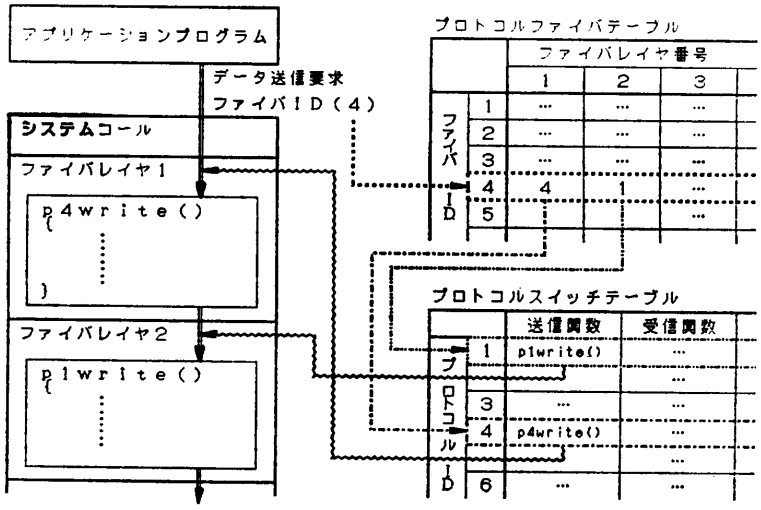

図2 プロトコルファイバとプロトコルスイッチ
Fig. 2 "Protocol Fiber" and "Protocol Switch."

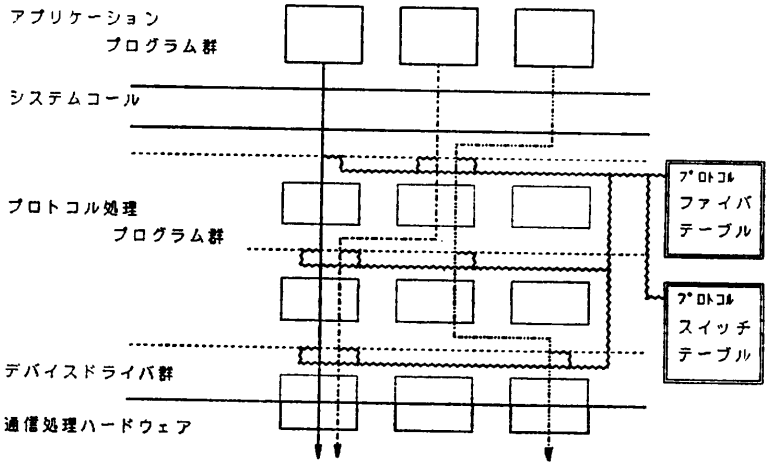

図3 プロトコル処理プログラム構成法
Fig. 3 Structure design of protocol modules.

5. おわりに

本論文では，プロトコル TCP/IP と X.25 を例として，異種プロトコルの実装手法について述べた．その特徴と効果を以下にまとめる．

(特徴) プロトコル TCP/IP，X.25 の整合を図るため，両者の間にレイヤを設け，インタフェースプロトコルを導入した．インタフェースプロトコルとしては，呼制御，アドレス変換を規定した．

(効果) 本手法により，プロトコル処理モジュール間の独立性を確保しつつ，プロトコルの整合が可能となり，性能は実用に耐えうる．また，ネットワークアプリケーションプログラムの通信路からの独立性を確保できる．

また，この実装手法を一般化して，プロトコルファイバ，プロトコルスイッチという概念を用いたプロトコル処理モジュール構成法を提案した．

謝辞 本研究を進めるにあたり，終始ご指導いただいた NTT 情報通信処理研究所応用システム研究部長石野福弥博士，大橋楷一郎グループリーダ，君島敏夫グループリーダに深謝致します．

参考文献

1) 竹内，工藤：DDX パケット交換網を介した TCP/IP の実現法，第 34 回情報処理学会全国大会論文集，3Z-1 (1987)．

2) 工藤，竹内ほか：DIPS における UNIX 制御方式，通研実報，Vol. 37, No. 2, pp. 167-173 (1988)．

(昭和 63 年 4 月 13 日受付)
(平成元年 2 月 14 日採録)

竹内 宏典 (正会員)

昭和 37 年生．昭和 59 年大阪府立大学工学部電気工学科卒業．同年日本電信電話公社 (現，NTT)・横須賀電気通信研究所入所．現在，NTT 情報通信処理研究所応用システム研究部研究主任．オペレーティングシステム，分散処理システムの研究に従事．

工藤 明彦 (正会員)

昭和 29 年生．昭和 52 年京都大学工学部電気工学科卒業．同年日本電信電話公社 (現，NTT) 入社．現在，NTT 情報通信処理研究所主任研究員．主に，DIPS オペレーティングシステム制御プログラム，分散処理システムの研究・開発に従事．電子情報通信学会，JUS 各会員．