

技術展望

UDC 681.324.078 : 006.72.022

プロトコル (通信規約)

田中英彦

田中英彦：正員 東京大学工学部電気工学科
Communication Protocol. By Hidehiko TANAKA, Regular Member (Faculty of Engineering, the University of Tokyo, Tokyo).
資料番号：昭 53-169 [技術展望-8]

1. はじめに

プロトコル (通信規約) という言葉が最近よく使われるようになったが、この言葉が現在のような意味あい使われ始めたのは大規模な計算機網 ARPA 網のプロジェクトが 1970 年の SJCC において発表されたからであろう。そこでは HOST/HOST PROTOCOL という言葉で「主計算機間の通信のための取定め」を指すものとして使われたが、それ以来この言葉は広く様々な通信し合う“実体” (プロセス, プロセッサ, OS 等) 相互間の諸取定めを表す言葉として使われるようになった⁽¹⁾。

この背景としてはもちろん、上述のように将来の情報処理ユーティリティを目指した計算機網の試行が広く行われ始めたことがある。ここでは複数の計算機が互いにインタラクトし合う形態をプロセス間の通信としてとらえ、それを実現するために各 OS 相互間の通信連絡上の約束がプロトコルという形で定められた。次に計算機と通信の結合によるオンラインシステムの発達がある。すなわち、複雑化するオンラインシステムに対しその構造を統一しようとするネットワークアーキテクチャの動きがそれで、これは 1974 年 9 月に IBM が発表した Systems Network Architecture を嚆矢(こうし)とするが、オンラインシステム全体の機能を明確にしそれを階層構造に整理して、各装置に置かれた対応する同階層相互間の通信と隣接階層間の通信とに分け、前者に対してはプロトコルを明確に定め

後者に対してはインタフェースを明確化するようになった。3 番目は公衆データ専用網の出現である。従来、データの伝送には電話回線を利用しデータ交換や伝送制御等は別の装置で行うのが普通であったのに対し、それらの機能のかなりを網側に備えることにより経済的で高品質なデータ通信システムを構成できることが期待されているものである。これに対しては当然、ユーザーと網との間のインタフェースが問題となり CCITT を中心として標準化が行われている。このインタフェースでは従来の電話回線インタフェースのように簡単なものだけに止まらず、より論理的なレベルまで定める必要がありプロトコルとして明確化されることになる。

これら三つの流れは互いに密接な関連を持っており、現在のところ、まだ完全に融合はしていないが、いずれより両立性のあるものになってゆくであろう。

プロトコルを定めることは各レベル層の機能を間接的に定めることとなり、その設定は通信し合う両者に矛盾のない形で行わねばならないが、それが正しいか否かの判定はプロトコルが複雑になればなる程難しく実装前にその正当性の検証手続きが必要になる。又、プロトコルの記述法、および実装法 (Implementation) も実際にシステムを作成する上で重要な問題である。

本文では、以上のようなプロトコルを巡る流れと、それらプロトコルの内容、および実現上の問題等について展望を行う。

2. 汎用計算機網のプロトコル

2.1 プロトコルの構造

計算機センター間網等、汎用計算機網として代表的なものには、ARPA 網を始めとしてフランスの CYCLADES、ヨーロッパの EIN、我が国の JIP-NET^{(14)~(16)}、大学間網 N1^{(11)~(13)}などがあるが、これらのシステムで設定されているプロトコルの構造はすべて、通信回線の制御手順を最下層とし様々な応用プログラム相互間の約束事を最上層に置きその間を基本的には3レベルの層に分けたものである。例えば図1は ARPA 網^{(9)~(10)}の例であるが、ミニコンピュータ (IMP) 相互を専用線で結んだデータ交換網 (サブネット) に各計算機システムが接続されている。プロトコルとしては、サブネットにおける隣接 IMP 相互間の制御用プロトコル (伝送制御手順、パケット形式、障害回復手順)、メッセージ発生源 IMP とあて先 IMP 相互間のプロトコルがある。後者はメッセージのパケット分解・組立、メッセージ ACK (RFNM)、バッファ予約等に関する取定めである。HOST 内におけるプロトコルとしては、メッセージを授受しあう HOST 相互間の HOST/HOST プロトコルとその上に設定され、様々な応用目的に使われるための機能向きプロトコルがある。後者は代表的な網の使い方に即したもので、幾つかのプロトコルが設定されておりユーザーはこれらを利用し、更にその上にユーザー独自の応用目的に合せた約束事を設定することになる。

CYCLADES や EIN で用いられているプロトコルの構造も同様な形^{(17)~(19)}であるが、HOST/HOST プロトコルに相当する Transport プロトコルの下部構造に特徴がある。すなわち、トランスポートステーショ

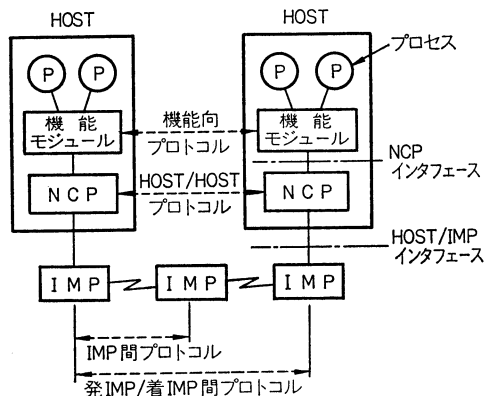


図1 ARPA 網のプロトコル構造

ン (HOST) 間のメッセージ授受をバーチャルコールとデータグラムの2階層構造としている点で、ステーション間の通信でメッセージの順序制御、誤り制御、フロー制御等を必要とするものに関してはポート間での制御を行うが、それは上層部で行い下層部では何らその制御を行わず、又、そのようなモード (datagram) で送られるデータも存在する。このデータグラムプロトコルは従って、パケットフォーマットとアドレッシングのみよりなる。更にこの下層のプロトコルは伝送制御手順でありそれには HDLC が用いられる。CYCLADES ではこの下層部のためのデータグラム網を CIGALE と呼んでいる。

大学間網 N1^{(11)~(13)} は公衆データ網 DDX を用いた計算機網で、IMP のような中継交換機能は不要で、HOST と DDX とを接続するためにフロントエンドプロセッサ (FEP) が置かれる。又、N1 の場合、DDX の回線交換とパケット交換の両機能を併用しており、それぞれに対して別個のプロトコルが設定されている。

2.2 HOST/HOST プロトコル^{(2),(3),(13),(15),(18)}

プロセス間通信をサポートするために、各プロセスが存在するシステム間でその制御用の取定めが設定される。この HOST/HOST プロトコルの内容は一般に、次のようなものである。

- (i) メッセージ形式
- (ii) プロセスの識別法、プロセス間リンクの設定
- (iii) メッセージのフロー制御/順序制御/誤り制御
- (iv) 初期化手順、テスト機能、割込機能

これらを両システム間で制御するためにそれ用のコマンド等が用意される。

代表的な三つの HOST/HOST プロトコル、ARPA 網 HOST/HOST プロトコル⁽³⁾、CYCLADES Transport プロトコル⁽¹⁸⁾、N1 HOST/HOST プロトコル⁽¹³⁾の比較を表1に示す。HOST/HOST 間の情報授受単位の長さは制御情報も含めたビット長であり、ARPA ではこの単位ごとに順次確認がとられる (RFNM による) が、CYCLADES 及び N1 ではこの転送単位に逐次番号がふられ、それを用いて集合 ACK やフロー制御が行われる。転送情報の発生源となったプロセスと受信プロセス及びそれらの情報出入口を指定するためのポート番号は HOST アドレスを除いた部分である。CYCLADES ではこのポートを複数のリエゾンに対して共有することが許されている

表 1 HOST/HOST プロトコルの比較

項目	ARPA 網	N1	CYCLADES
プロトコル名	HOST/HOST	HOST/HOST	End-to-End
主システム名	HOST	HOST	Transport Station
NCP 間授受単位 (ビット)	メッセージ 8095	セグメント 2040	レター/テレグラム 2112/16
単位識別番号	なし	6 bits	8 bits
ヘッダ長 (ビット)	72	80	40~可変
メッセージ最大長 (ビット)	8023	1960×64	2048×128
ポート名	ソケット	論理ポート	ポート
ビット長	32	32	16
ポートの共有	不可	不可	可
プロセスとの対応付け	動的	動的	固定
リンク名	リンク	論理リンク	リエゾン
方向性	単向	単向	双方向
バーチャルコール / データグラム	VCのみ	VCのみ	VC DG並用
誤り制御	メッセージごと	一連セグメントの集合 ACK	一連レターに対する集合 ACK
フロー制御	RFNM とバッファ割当	SFC によるウィンドウ方式	クレジットによるウィンドウ方式
最大ウィンドウサイズ	—	255	15
送信側からのバッファ要求可	不可	可	不可

ほか、ポート番号と各プロセスとの対応関係は固定的であるが、ARPA, N1 では動的に定められる。

又、CYCLADES では情報の転送モードにレターとテレグラムの2種類があり、レターは ARPA/N1 のメッセージと同種のものであるがテレグラムは16 bits 固定長の情報であり、応答確認やフロー制御が行われない。これはユーザーインタフェースには一つのイベントとして見え、割込み等に使用される。

フロー制御に関しては、ARPN 網では RFNM による順次制御とバッファ割当てを用いているが、N1 では SFC/RSC コマンドによるウィンドウ方式を、CYCLADES では Credit の割当てによるウィンドウ方式が使われている。

2.3 機能向きプロトコル

HOST/HOST プロトコルの設定によって一応各 HOST 内プロセス相互間の通信が可能となるが、これだけではユーザーにとって使い難くより利用目的に沿った様々なプロトコルが設定される。すなわち、代表的な網利用形態の幾つかに対して網内での標準利用形態を設定し、各 HOST 内ではそれとの間の変換を行うことによって異なる OS や機種を備えたシステムの相互利用を容易にすると同時に、標準的な使い方の幾つかをシステム側で用意しておくことによってユーザーの便宜を図ろうとするものである。この代表的なプロトコルには以下のようなものがある。

(1) ICP (初期接続プロトコル)⁽⁴⁾

これはユーザーのプロセスと、利用目的のプロセスとを最初に接続させるための手続きで、一般には相手のポート番号が不明であるため必要となる。すなわち、特定のポート番号を定めておき、まずそこへ接続することによって相手ポート番号を知る。

(2) NVT (網仮想端末)⁽⁵⁾,
^{(35), (40)}

網内のすべての端末から全システムの会話形利用ができるためには、会話形端末として標準の端末を設定し、各 HOST ではローカルな端末のインタフェースとの変換をする必要がある。このような標準端末を Network Virtual Terminal と呼ぶ。NVT として

はキーボードとプリンタを備えたような端末が設定され、基本機能とオプションの設定、コードセットの標準化、エコーの取扱い等が問題になる。

(3) 会話形利用プロトコル^{(5), (19), (40)}

ARPA では Telnet と呼んでいるものがこれに相当するが、NVT と同意義で使われることもある。その中味は、ICP, NVT の利用および制御コマンドよりなる。コマンドとしては、端末からのブレイク信号の通知、相手プロセスへの割込み、情報転送と端末操作の同期、端末オプションの協議用等が設けられる。

(4) RJE プロトコル^{(9), (12), (17), (40)}

リモートバッチのためのプロトコルがこれで、その内容は RJE 用コマンド/応答の設定、ソースデータ/出力結果の指定法、形式などからなる。ARPA では図2のような構造をとり、コマンド/応答を授受するリンクとデータを授受するリンクとが独立に設定

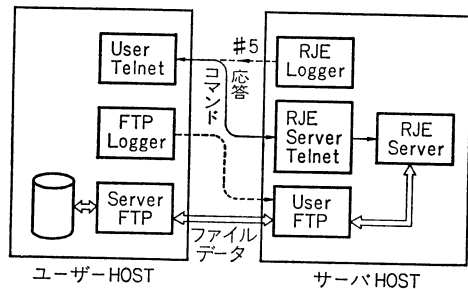


図 2 ARPA RJE プロトコルの構造

され、前者には Telnet コネクションを用い後者には後述のファイル転送プロトコルを用いるのが標準になっている。しかし、N1 では RJE 用には一対のリンクしか用いず、その上でコマンド/応答や入出力データを混せて授受しあう形をとっている。

(5) ファイル転送プロトコル (FTP)⁽¹⁷⁾

諸システム間のファイル共有を可能とするために不可欠のプロトコルで、計算機網の利用が発達すれば非常に重要となることが予想される。しかし、ファイルの構造、コマンド、制御システムは各 HOST によって異なる部分が多く、ファイルシステム全般にわたって統一的な標準プロトコルを設定することは難しく、一般に設定されるファイル構造としてはレコード構造をしたシーケンシャルファイル程度であり、データ表現タイプも非常に限定される。

(6) コマンド/応答プロトコル⁽²⁰⁾

前述のように機能向きプロトコルでは、ICP を除きコマンドを送り応答を受けるという形がすべてに共通している。従って、HOST/HOST プロトコルと機能向きプロトコルの間に、コマンドを送り応答を受ける標準機構が用意されていればあらゆる応用に使うことができ便利であろう。これにより、リモートリソースをファンクションのレベルで利用できることになるので網リソースとして新しく様々な機能を追加するのが容易となる可能性がある。

(7) 網ジョブ制御プロトコル⁽²¹⁾

より密接なシステム間共同作業を行うためのプロトコルとして、各 HOST それぞれで動いているプロセスの集合でもって一つのユーザージョブとみなしそれらの間の制御を行うためのプロトコルが考えられる。この研究の一例を図3に示す。これは各ジョブの制御を各計算機ごとの管理とそれを総括する管理とに分けて行うもので、それら両管理プロセス間の制御が網内のプロトコルになる。

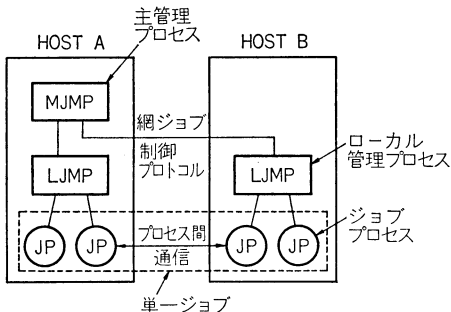


図3 網ジョブ管理システム

(8) その他の機能向きプロトコル

以上のほか、各所で検討されたものとしては、グラフィック端末用プロトコル⁽¹⁹⁾、メールボックスプロトコル⁽⁸⁾、データ再編成プロトコル⁽⁹⁾、データベースアクセスプロトコル等がある。

2.4 ゲートウェイ

複数の網間を接続する場合、各網内で設定されているプロトコル体系の差が問題となる。このような場合、網間に Gate Way と呼ばれるインタフェース装置を設けプロトコル相互の変換が行われる。

3. ネットワークアーキテクチャ

3.1 ネットワークアーキテクチャの概要

IBM がオンラインシステムの統一的なアーキテクチャとして SNA を発表して以来、各社より続々とネットワークアーキテクチャの発表が相次いで行われた。この背景をなすものは、通信アクセス法や伝送制御手順の多様化によりシステムの変更・拡張が困難になってきたことと、一方では分散処理技術の発達、情報処理の広域化、高度化に伴いオンラインシステムとしての統一的なアーキテクチャを設定して諸機能を整理し体系化する機が熟したと見ることもできよう。

従って、その目的は、網構造・端末の種類と応用プログラムとの独立性を高め、回線・端末・回線制御プログラム等を複数の応用プログラムで共用すると共にシステムの可用性を向上させることがあげられている。ここでは、代表的であると共に対照的なネットワークアーキテクチャである IBM SNA と DEC DNA を取上げ、その構造を簡単に述べる。

3.2 IBM SNA^{(22)~(25)}

SNA の制御システムの目的は、2 エンドユーザー（プログラムや端末オペレータ）間の情報交換手段を提供することにあるが、そのために網内の論理的ポートとして Network Addressable Unit を設け、網内のデータや制御情報はすべてこの NAU 間を走る。NAU 間の論理的リンクはセッションと呼ばれ網内の管理はすべてこのセッション確立後に行われる。

(1) 構造

図4にノード内の構造を示す。DLC はデータリンク制御部でこれには SDLC を用いることを基本としている。PC はパス制御で各メッセージのアドレスを基にあて先の NAU へ情報を届けるための径路制御を行う。TC は伝送制御部で、NAU 間のデータ伝送管理や通番管理、セッションの開始/終了などを担当す

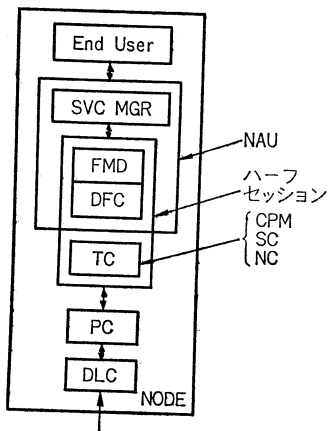


図4 SNAにおけるノード構造

る。以上の3部分を合せて伝送サブシステムという。

DFCはデータフロー制御部で、FMDは機能管理部であり、アプリケーションからのリクエストを特定のデータ形式に変換したり出力装置選択のようなシーケンス制御を行う部分である。SVC MGRはサービスマネージャでエンドユーザー(EU)の利用形態に合わせるためのインタフェースである。以上のうち、DFCからSVC MGRの部分は機能管理層と呼ばれNAUを構成する要素である。

(2) メッセージ形式

ノード間を流れるメッセージの基本形は図5に示すようなものである。DLCヘッダとトレーラは伝送制御手順の部分で、それ以外の部分はBTUと呼びヘッダ(TH)が付く、そのテキスト部をBIUと呼ぶが中味はヘッダ(RH)とRequest/Response Unitに分かれる。THはPC部が主として扱う部分であって先/発元アドレス、逐次番号、その他からなり、RHはRequest/Responseの区別、RUの Kategorie (あて先、FMD/NC/SC/DFC あて)のほか、伝送の区切り、フロー制御等のフラグからなる。コマンドの内容としてはセッションを張ったり切ったりするセッション制御、データトラヒック制御、フロー制御、網制御、端末制御などがある。

(3) SNAの利用

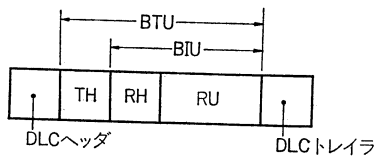


図5 SNAのメッセージ形式

SNAは単一HOSTの端末網システムが中心であるが複数HOSTをも含むように拡張(MNF)されている。CCITT X.25との整合については、3705通信制御装置用アダプタを用いて整合を取ることが公表された。

SNAのHOST内の実現はVTAMであるが、応用システムであるIMS/VS, CICS/VS, RJE (JES 2/3, RES)等はすべてVTAMの上に作られており、高位レベルプロトコルとしてRJE用RTAM等がある。

3.3 DEC DNA⁽²⁶⁾⁻⁽²⁸⁾

(1) 概要

DECのDigital Network Architectureは1975年5月に発表されたがこのアーキテクチャはSNAとは対照的で、ミニコンピュータ網指向のソフトウェア体系である。すなわち、複数のコンピュータを統一的思想のもとに接続し網内にあるプログラム、デバイス、ファイル等の共有を可能にすると共に、複数の計算機それぞれの中で動くプログラムの共同作業をも可能とすることを目的としている。従って、DNAのソフトウェア体系はユーザーに汎用のリソース共有手段を提供するためのものであって、SNAに比べて単純ではあるがopen-endedな設計となっている。

(2) 基本構造

コンピュータシステム内の諸リソースをオブジェクトの集合とみなし、これらオブジェクト間の通信機構を提供してオブジェクトの共有手段を提供することがDNAの目的であり、二つのオブジェクト間に構成される仮想的な通信路を論理リンクと呼ぶ。DNAの基本構造は図6に示すように4階の階層をなしている。

Dialogue Levelは論理リンクを介して情報の

授受を実際に行うレベルで、ユーザープログラム間での情報授受のほか、ファイル、入出力装置、端末などを対象にしたData Access Protocolも用意されている。

Logical Link Levelは論理リンクの確立/切断を行うレベルで、ノード間情報のルーティング機能も持つ

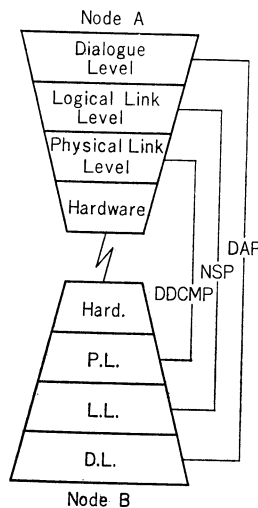


図6 DNAの基本構造

ており Network Service Protocol が設定されている。

Physical Link Level は伝送制御手順をサポートする。このレベルのプロトコルは Digital Data Communication Message Protocol と呼ばれている。

Hardware Level は回線とビットストリームの授受をするためのドライバ/レシーバ部分である。

(3) プロトコルの概要

• DDCMP：フレーム構成は SOH/ENQ 等の制御文字を先頭にしてフレーム長情報と CRC コード等が付き、フレームに逐次番号が付けられていて HDLC と同種の誤り制御、順序制御を行う。

• NSP⁽²⁷⁾：このサポートするリンクモードには三つありデータグラムに相当するものとリンクを設定してからデータ授受を行うモードがあるが、後者はデータのフロー制御、ACK 確認、順序制御をするか否かでも分けられている。

• DAP⁽²⁸⁾：これは NSP を用いる高位レベルプロトコルで、リモートデバイスからの入出力、リモートファイルへのアクセス等を簡便に行うためのものである。

• MOP：通信リンクを介して他のコンピュータにプログラムをロードしたりメモリダンプやテストをするためのプロトコルである。

4. 公衆網インタフェースと プロトコルの標準化

4.1 公衆データ網

新データ網にユーザー端末を接続する場合のインタフェースは図7に示すとおりである。データ網業者の管理範囲は DCE (Data Circuit Termination Equipment) までであり、ユーザーの装置 DTE (Data Terminal Equipment：端末や計算機など) がそれに接続されて網にインタフェースする。この場合 DCE-DTE インタフェースがいわゆる公衆網インタフェースである。公衆データ網のサービスとしては一般に、回線サービス(データ専用線)、回線交換サービス、パケット交換サービスがあり、これらに対してそれぞれのインタフェースが用意される。

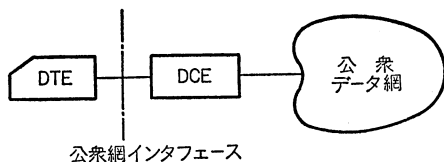


図7 公衆網インタフェース

4.2 公衆網インタフェースの標準化

新データ網に対する国際的な諮問機関は CCITT の第7研究委員会 (SG VII) であるが、前会期末の総会 (1976年9月) において新データ網に関する17件の X シリーズ勧告が制定された。そのうち、データ専用線や回線交換に対する勧告の規定範囲は、コネクタ、インピーダンス等の物理的電氣的条件と回線交換の場合はダイヤル信号送出規準を定めているに止どまりそれ以上の伝送内容に関しては透明である。これに対し、パケット交換網に対する勧告はより高位の論理レベルに対する規定をも含んでいる^{(29),(30),(32)}。

(1) パケット形態端末のインタフェース条件

この勧告 X.25 は次の3レベルの規定からなる。

- ① 電氣的・物理的インタフェース条件
- ② リンクレベルプロトコル
- ③ パケットレベルプロトコル

① としては回線交換(同期式)と同じ X.21 又は X.21 bis が適用され、② に対してはハイレベル伝送制御手順 (HDLC)⁽³⁰⁾ が適用される。HDLC に関しては ISO でも国際標準が定められていたが両者の間に食違いを生じたため 1977年4月に両者の合同会議が開かれ両者の統一が図られるところとなった。③ は図8のようにパケット内容に対する規定で、論理チャネルの設定や切断のための制御、更にウインドウ方式によるパケットの流量制御も行う。

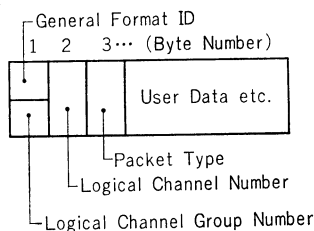


図8 パケットフォーマットの一例

(2) PAD インタフェース

パケット網においては、一般端末を網に接続するためには端末からのメッセージをパケット化したり、又、網からのパケットを元のメッセージ形に変換するための装置が必要である。これを一般に PAD (Packet Assembly Deassembly) と呼んでいるが、PAD を用いる場合のインタフェースも暫定勧告が制定されており、対象とする端末は当面 110, 200 又は 300 bit/s の速度を持つ調歩式端末である。

ところで、勧告 X.25 はもともと、Virtual Call を

もとに作られているが、データ通信網の基本サービスとしてそれが最適かどうかは議論のあるところであり、網側の機能をより簡略化した形のデータグラムをサポートするべきであるという意見もあり⁽³¹⁾、将来の課題となっている。

4.3 トランスポートプロトコル

公衆網インタフェースとして勧告が出されているのは DTE-DCE インタフェースであって、網を介して接続されている両端の装置間の規約ではない。この End-to-End レベルのプロトコルはそれらの上に設定されるべきものであり機能からいえば前述の HOST/HOST プロトコルに相当する。又、複数の異なる計算機網どうしを相互に結合してできたグローバルな網における End-to-End のプロトコルの必要性も考えられる。この標準案として提案されているのが IFIP Internetwork Working Group (WG 6.1) より出されている Transport プロトコル⁽³²⁾である。このプロトコルは CYCLADES, EIN 等で用いられているプロトコルを基礎としており、ISO でもこの種のプロトコルの標準化作業が進められている。

(1) 構造

コンピュータシステムを一般化した Virtual Installation が通信機能によって相互に結合されたシステムを考える。各 Virtual Installation には通信機能をサポートする部分が置かれそれを Transport Station(TS) と呼ぶ。従って、通信し合う実体はこの TS の Subscriber (SB) と呼ばれ TS サービスを受けるが、TS は TS 相互間のトランスポートプロトコルに従って動作する。又、各 SB はポートと呼ぶサブアドレスの集合を持ちこれを通してデータの授受を行う。

(2) 基本トランスポートサービス

TS が提供する基本サービスは、フローの上で両方向にレターとテレグラムとを SB 間で交換することである。レターはデータ転送用でバイト(8 bits)を単位とした可変長データであり、テレグラムは2バイト固定長の情報で割込みや制御用情報として用いる。この基本サービスは各レター(テレグラム)ベースで行われ、初期の同期や準備は不要である。

(3) 付加サービス

基本サービスのほかに、SB からの要求に応じてフローごとに他の付加サービスが行われる。これは、ある期間(セッション)を単位として行われ、レターに対する誤り制御、フロー制御、短縮アドレス等があ

る。このセッションは両 TS 間の同意に基づいて始まりどちらかの TS からの要請に応じて終了する。

誤り制御に関しては、レターに付けられる逐次番号を用い HDLC の要領で行われる。フロー制御は送信側 TS に対してそのフロー上で送信可能なレター数を示す credit と逐次番号とを用いて行う。

短縮アドレスに関しては、セッションの開始時にそのフローに対するプラグ対番号(8 bits ずつ)を定めて互いに通知し合い、フロー ID の代りにプラグ番号を用いる。

4.4 ネットワークアーキテクチャの標準化

現在までに標準・勧告が出されている部分はいずれも通信回線に近いレベルのもののみであるが、より広範囲にネットワークアーキテクチャという概念に沿って各システムの OS を作り制御を行うには、システム全体の機能を洗い直して構造を明確化しさまざまな高位レベルのプロトコルを定める必要がある。これについては ISO の SC 6 プロジェクト 24 で検討が進められてきたが、昨年3月の ISO/TC 97 総会の決議として以上の作業を更に積極的に進めるべく SC 16 の設置が定められた。

この作業タイトルは Open System Interconnection であり、1978 年2月末、第1回の SC 16 会議が開かれた。そこで早速、ネットワークアーキテクチャの暫定モデルがドキュメント N 34⁽³⁴⁾ として出され、今後の進め方としてはワーキンググループが三つ設立されることとなり、具体的な作業が進められている。

(1) 暫定モデル

これは図9のような7層からなるものであって、そのうち、下層4層は X. 25 等で定められているものなどに対応するが、上4層については次のような機能を持つものとされている。

① レベル4 : Transport End-to-End Control

7	Process Control
6	Presentation Control
5	Session Control
4	Transport End-to-End Control
3	Network Control
2	Link Control
1	Physical Control

図9 SC 16 ネットワークアーキテクチャの暫7レベルモデル

Level, ネットワーク機能や形態の複雑さに関わらず, ネットワークサービスユーザーに同一のインタフェースを見せるためのレベル.

- ② レベル5 : Session Control Level, プロセス間のダイアログ機能を提供するレベル.
- ③ レベル6 : Presentation Control Level, 転送される情報に対して必要な変換を施すレベル. 例えば, 暗号化, 圧縮, NVT 変換, データ形式・コード等の変換.
- ④ レベル7 : Process Control Level, 情報処理機能としてアプリケーション及びシステムに関する動作を行うレベル. 例えばシステムレベルでは, RJE, FTP などのプロトコル, 端末制御などがある.

このモデルを見るとき, レベル4として, 複数の網が結合された場合にも End-to-End のプロトコルを一意的に定めようとしている点が重要で, ここを境にして網の完全な仮想化が行われており複数の網間結合が重要となるであろう将来を見越した設定である.

(2) 諸機能向きプロトコルの標準化

上記アーキテクチャは, 網のさまざまな利用形態に即した機能向きプロトコルの設定によって現実化される. そのようなプロトコルとして当面考えられているものは, RJE を含むプロセス間結合とセッション管理, 網仮想端末 (NVT), 標準データの交換 (FTP) 等である.

5. プロトコルの記述と実装^{(37)~(39)}

5.1 プロトコルの設計問題

プロトコルは一般に通信し合う2者間の規約であるが, その設計に当たっては, 正常な動作状態においては能率高く動作すると共に時折生じるエラー状態やサブシステム障害時に対処して, 前者のエラーに対しては再試行による回復を試み後者に対してはシステムの再構成に入る等の考慮が必要である. 又, 一般にプロトコルの正当性を証明することはシミュレーションやテスト等によって行われることが多いが, 論理的に正当性を証明できればシミュレーション等によっては見出すことが難しいプロトコルの弱点やエラーを検出する良い道具となる.

5.2 プロトコルの記述

設計したプロトコルは何らかの形で記述する必要があるが, その記述法は一般に二つに分かれる. 一つは通信し合う両者の動作を同時に記述するもので, 他は

片側の動作のみに着目して記述する方法である. 前者はそのプロトコル全体の動作をは握する上で都合が良いが, 複雑となりやすく, 又, 実装とは直接関連を持たせ難い. 後者は実装に都合が良いが全体としての動作をとらえ難く, 特殊な状況の発生は握が難しい. 一般にプロトコル記述に望ましい事柄としては,

- ① プロトコルを抽象レベルの異なった形に分割し, 下位レベルを抽象データタイプとして, より分かりやすい形で定義できること.
- ② 記述が正当性の証明に向いていること.
- ③ 実装が直接的かつ単純で, 部分的にせよ自動化ができること.

等であり, これらをすべて兼ね備えた記述法は現在のところ開発されていないが, 一般に使われているものをあげると, PL/1 等の高級言語を用いる方法のほか, 動作状態と遷移事象による行列表を作り, その中で動きを記述する動作マトリクス法, システム全体を有限状態マシンの集合としてとらえ, それらを個々に情報の流れと動作の流れを合わせて記述する方法, および競合関係の表現に重点を置いた Petri-net などがある.

5.3 プロトコルの実装

実装に関しては通常のソフトウェアとあまり変りはないが, 特に, プロトコルを階層化し抽象化を行って注目すべき事象範囲を限定し他は black box とみなすという構造化プログラミングの手法は有効であろう. 更に, 高級言語を用いれば, そのプログラミング労力の減少, 誤りの発見に都合がよく, 又, そのプログラムを他のシステムに移殖する場合の移植タビリティが高くなり文書化にも適することとなる.

又, プロトコルの動作を同時動作する複数のプロセスの集合として実装化することができればその実装はかなり容易になることが考えられ, Concurrent PASCAL, PL/1 等, 並列プロセス動作を能率良く記述できると共にその実行効率の高い言語の開発が望まれる. 更に, 様々なプロトコルが作られ実装されてゆく過程では, 当然そのプロトコル自体の評価法が問題になろう.

6. おわりに

現在はデータ通信とコンピュータの融合の時期にきており, 今後ともますますその関係は密接になるものと思われる. その時期にあって, ネットワークアーキテクチャを定め標準化してゆくことは発展の上からも

是非早急に実現せねばならないことである。X. 25 の勧告が出されて以来ハイレベルプロトコルの標準化が急に話題に上り国際的にもその作業が急がれている時期であるが、一方、この種の標準化はその影響がじん大で、一度定められてその製品が広く出回ってしまうと、もはやその変更は非常に困難となる。将来を見越したより発展性に富むアーキテクチャの実現を望みたい。

文 献

- (1) 田中英彦：“計算機網のソフトウェア”，信学誌，58，11，p. 147 (昭 50-11).
- (2) 伊藤哲史：“HOST-HOST プロトコルおよびハイレベル・プロトコルの動向”，情報処理，18，9，p. 925 (昭 52-9).
- (3) “HOST/HOST protocol for the ARPA network”，NIC 8246.
- (4) “Initial connection protocols”，NIC 7101，7155，7103.
- (5) “TELNET protocols”，NIC 18639，18640，15389.
- (6) “Remote job entry protocols”，NIC 12112.
- (7) “Fire transfer protocols”，NIC 17759，30843.
- (8) “Mail protocol”，NIC 24664.
- (9) “Data configuration protocols”，NIC 6780.
- (10) “Network graphics protocols”，NIC 24308.
- (11) 浅野，田畑：“N-1 プロジェクトにおけるネットワークプロトコルの基本構造”，第 16 回情報処理学会大会，3 (昭 50-11).
- (12) 北川，金沢，長谷部：“N-1 プロジェクトにおける RJE プロトコル”，同上，4.
- (13) 田中，浅野，長谷部，田畑：“N-1 プロジェクトにおける HOST/HOST プロトコル”，同上，5.
- (14) 山本：“コンピュータネットワークシステム JIPNET”，同上，108.
- (15) 伊藤，鍛冶，西原：“JIPNET の HOST-HOST プロトコル”，同上，109.
- (16) 小川，伊藤，鍛冶：“JIPNET における高位プロトコル”，同上，114.
- (17) Pouzin, L. : “CIGALE, The packet switching machine of the CYCLADES computer network”, Information Processing 74, North-Holland Pub. Co. p. 155.
- (18) Zimmermann, H. : “The CYCLADES end-to-end protocol”, 4th Data Comm. Symp., Québec (Oct. 1975).
- (19) Zimmermann, H. : “Terminal access in CYCLADES computer network”, Reseau CYCLADES Ter # 510 (1974).
- (20) White, J.E. : “A high-level framework for network-based resource sharing”, NCC 1976, p. 561.
- (21) 田中，山内：“ネットワークジョブ管理システム”，第 17 回情報処理学会大会，266 (昭 52-11).
- (22) McFadyen, J.H. : “Systems network architecture; An overview”, IBM Syst. J., 15, 1, p. 4 (1976).
- (23) Cullum, P.G. : “The transmission subsystem in systems network architecture”, *ibid.*, p. 24.
- (24) “複数システム・ネットワーク機能”，IBM マニュアル，N; G 518-5067.
- (25) “Systems network architecture format and protocol reference manual; Architecture logic”，IBM Manual, SC 30-3112-0.
- (26) Conant, G.E. and Wecher, S. : “DNA—an architecture for heterogeneous computer networks”, 3rd Int. Conf. on Computer Communication, Toronto (Aug. 1976).
- (27) “Digital network architecture, specification for network services protocol NSP”, DEC (Nov. 1976).
- (28) “Digital network architecture, design specification for data access protocol DAP”, DEC (July 1975).
- (29) 石野福彌：“パケット交換網の通信規約”，情報処理，18，11，p. 1148 (昭 52-11).
- (30) ISO Document, IS 3309, DIS 4335, N 1385, DIS 6159, N. 1444.
- (31) Pouzin, L. : “Virtual circuit vs. datagram-technical and political problems”, NCC 1976, p. 483.
- (32) CCITT Document AP IV-No. 54-E, Final Report on the work of study group VII, Geneva (1976).
- (33) Zimmermann, H. and Elie, M. : “Transport protocol, standard end-to-end protocol for heterogeneous computer network”, Reseau CYCLADES, SCH 519.2, (May 1975). (revised version of INWG note 61).
- (34) ISO/TC 97/SC 16 N 34, International Standard Organization, Provisional model of open-systems architecture.
- (35) “Interactive terminal interface specification”, Telenet Comm. Co. (Sept. 1975).
- (36) “Host interface specification”, Telenet Comm. Co. (Nov. 1975).
- (37) Bochmann, G.V. : “Logical verification and implementation of PROTOCOLS”, 4th Data Comm. Symp., Quebec (Oct. 1975).
- (38) 浦野，小野，鈴木：“プロトコルマシンにおけるプロトコルとその記述について”，第 18 回情報処理学会大会，297 (昭 52-11).
- (39) 勝又 裕：“インタラクティブオートマタシステムモデルとそれを用いた相互交信系の設計法”，東京大学博士論文 (昭 50-03).
- (40) “OS 7 コンピュータネットワーク”，HITAC プログラムマニュアル，8700-3-31.