

2.3 計算機網のソフトウェア

田中英彦

田中英彦：正員 東京大学工学部電気工学科
 Software System of Computer Networks. By Hidehiko TANAKA,
 Regular Member (Faculty of Engineering, the University of
 Tokyo, Tokyo).
 資料番号：昭 50-180 [特集-33]

1. はじめに

複数の計算機システムをデータ通信回線やチャンネル結合などによって相互に結合して、1つの大きな処理系を構成した電子計算機網 (Computer Networks) は、1970年の始めごろから注目を浴び、現在各国において様々なシステムができ、又、計画されている。計算機網の目的は、一般に、計算機ハードウェア、諸ソフトウェア、データなど広い意味での資源 (Resource) を、網のユーザ間で互いに共有し共同利用しあうことである。これは、資源共有という言葉で呼ばれているが、その意味するところは広く、網それぞれによって重点の置きどころが異なる。これら計算機網の目的・実例などについては本誌上の解説⁽²⁾で既に取り上げられているし、一般的な構成技術についてもネットワーク特集号⁽¹⁾で解説されている。従って、本文では、別の観点からの計算機網について述べる。

すなわち、計算機網における通信を具体化するためのソフトウェアの問題点と役割および構成法についてである。計算機網における通信には幾つかの階層を持った意味がある。最も低いレベルでは、計算機システム間のデータ通信技術を意味し、その上には、異なる計算機システムそれぞれの内部における処理の基本単位 (プロセス) 間で適当な長さのビットストリングのやり取りを行うレベルがあり、更にこの上には、このメッセージが意味ある情報として相手に伝わるという意味での通信がある。この第3段階の通信が可能であることによって始めてその目的が達成されるわけであるが、計算機網では、特に後二者の意味での通信技術が問題になる。すなわち、計算機種の差異やオペレーティングシステムの差異があるため、これらを如何に

克服するかが重要な関心事である。

最も低い階層での通信はデータ伝送・交換の技術であるが、計算機網におけるデータ通信は、ユーザ端末と計算機間の通信と、計算機システム相互間の通信とがあり、特に後者は計算機網の特徴をなすもので、通常の端末通信とは異なり次のような特性が要求されることが多い。

- ① 高速大容量伝送
- ② メッセージ伝達遅延が短いこと
- ③ コードトランスペアレンシ
- ④ 対等な対向通信 (1次局同送)

①はファイル転送時に特に要求されることで、②は会話形処理を遠隔計算機の端末から行う場合の応答性からの要求である。又、③は、伝送されるデータがISO 7単位符号などで表されるものに止まらず、計算機内部の生のデータやコード化されたデータなどのビットストリング伝送が行われる場合に必要となる特性である。計算機間の通信系は、距離が長くなると回線接続が使われるのは当然であるが、上のような特徴もあって、比較的距離が短い場合にはチャンネル間直接結合とか、ディスク共有のような形で接続されることが多い。

これらのデータ通信については本特集の他のところで詳しく取扱われているのでここではこれ以上述べない。従って、ここで取扱う通信はそれより上の階層の通信である。

2. 計算機網におけるソフトウェアの問題点⁽²³⁾

計算機網における主要な問題は、いわゆる両立性の問題である。すなわち、一般に、オペレーティングシ

システムや言語、ソフトウェアなど、アーキテクチャの異なった計算機同志を如何にして協調させるかという問題である。もちろん、同種計算機の結合網であればこれらは比較的容易であるし、異種計算機網では両立性が全面的に問題となる。しかし、その差はあれ、どんな網であってもこれらの問題を全く避けて通ることはできない。これらには次のような問題がある。

① ハードウェア

プロセッサ構造の差、語長、命令、コード体系、また接続される周辺機器の差異などで、オブジェクトプログラムを直接計算機間で授受することはもとより、単なるビットストリングを授受する場合にも語長の違いが問題になる。プロセッサ構造の両立性に対しては、ダイナミックマイクロプログラミングによるエミュレータも1つの対策であろう。

② オペレーティングシステム

OSの機能は様々であり、又、多くのマクロライブラリやカタログした手順を持っている。計算機網において、これら様々な機能を能率良く統合させることは非常に難しい。

③ ジョブ制御

OSの提供する機能を表現するジョブ制御言語の差も問題である。

④ 言語

機械語に止まらず、FORTRAN、COBOLなどの高級言語においても計算機によって異なっているのが現状である。ある計算機に使われている高級言語を、他の計算機上で動くよう自動的に変換することは難しい。又、言語と計算機構造の親和性には差があり、ある計算機上では能率的な言語が、他では非常に能率が悪いことがある。

⑤ データ構造

データを貯えるのには多くの形式がある。例えば、順編成、索引順編成、その他、トリー形、リスト形など多くのファイル編成法がある。これらのファイルが網上で様々なディスク(容量やサイズの異なる)に分散し量が増してくると、データの直接的な転送が非常に難しくなる。

3. 計算機間通信

3.1 計算機間通信制御モニタの基本構造⁽⁵⁾

一般に、計算機間を結合するというこ

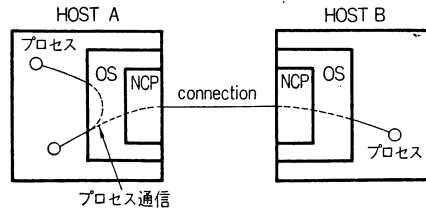


図1 プロセス間通信

は、処理系から見れば互いの計算機内で動いているプロセス間で情報を授受しあうということである⁽⁶⁾。すなわち、単一の計算機システム内でマルチプログラミングにより複数のプロセスが協調し合っていることを、そのまま複数の計算機にまで拡張したものと見ることができる。従って、計算機間通信を可能にするということは、互いの計算機内プロセス間の通信手段を設けることに他ならない。

プロセス間通信機構を設定する場合、計算機システムそれぞれの処理形態は一般に異なるので、異種計算機の網である ARPANET のような網では、計算機間で共通の取り定めを設定し各計算機 (HOST) がそれに従うという形を取る。この取定めをプロトコール (protocol) と呼んでいるが、それには様々な種類のものである。すなわち、プロトコールは図2のような階層構造で考えることができる⁽³⁾。ハイレベルプロトコールは実際の処理目的に密接に関連し、低レベルプロトコールは基本通信機構を受け持つ。中でも最も低位のプロトコールはメッセージを発信計算機から宛先計算機まで正しく届けるための諸取定めで、このレベルではメッセージの意味について何の設定もされておらず、単にある長さのビットストリングとし

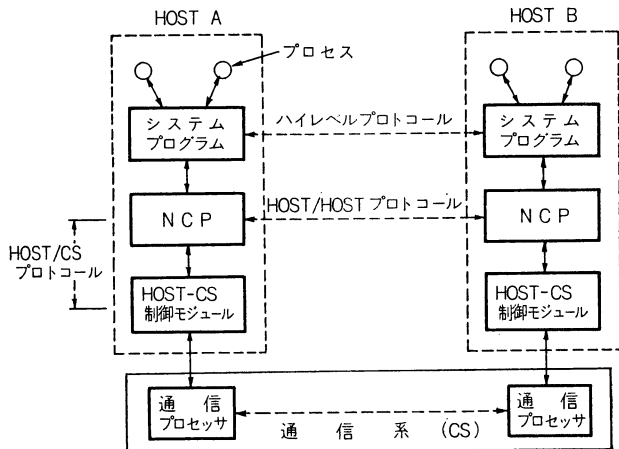


図2 プロトコールの階層構成

て扱われる。このプロトコルには、HOST と通信系 (CS) との間の取定めである HOST/CS プロトコル (ARPA では HOST/IMP) とか通信系内部のデータ通信の取定めなどが含まれる。各 HOST には通信系から見た計算機の OS である NCP (Network Control Program) が置かれており、異 HOST 内 NCP との間の共通した取定めである HOST/HOST プロトコルの下で NCP 間の連絡を行い、通常のデータメッセージの流れを制御する。

これまでのレベルのプロトコル構成によって、ユーザプロセスに便利な通信の基本手段を提供する。これによりユーザプロセスは、OS や通信の詳細を知る必要がなくなる。このレベルのインタフェースの上に様々な機能に向けたプロトコルがあり、計算機網を便利に利用するためのモジュール集合が存在する。

3.2 リモート・プロセス間通信⁽⁷⁾⁻⁽⁹⁾

遠隔プロセス間の通信には、次の2種がある。

- ① 接続向きプロトコル
- ② メッセージ向きプロトコル

①は、ARPANET⁽⁸⁾、JIPNET、INCS⁽²⁷⁾ など多くの網で用いられているもので、プロセス間情報の転送に先立ち、両者間で通信の契約関係を設定 (connection) してそれに識別番号を付与し、送られる情報にはその番号を付けておくことで識別をする形の通信形態である。従って、通常の電話のように相手のところまでの回線 (仮想的な) を捕捉し、それを用いて一連のメッセージ転送を行う。

②は D.C. Walden によって提案されたもの⁽⁷⁾で、毎回のメッセージ伝送ごとに相互プロセス間で送受の確認をし (データ送信要求の提示と受信側のバッファ準備完了)、確認が取れるとメッセージを転送する。それが済むとプロセス間の一時契約は解消する。①の方法では実際の通信制御を NCP に依頼して間接的に行っているのに対し、この方法は当のプロセスが直接通信に関与する点でより隔通性の高いプロセス間通信になっている。しかし、メッセージごとのオーバーヘッドは大きい。

①において、NCP 間で交わされるコマンドには、接続の確立・解消、フロー制御、割込み要求、NCP 制御ブロックの初期化、テスト、エラー通知などのレポートリがある⁽⁸⁾。これに対し②では、メッセージの送出要求、受諾通知、受諾拒否などのコマンドレポートリになる。

接続向きプロトコルにおいて、相手プロセスを指

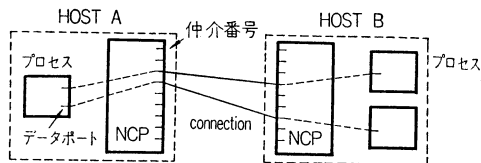


図3 標準仲介番号とコネクション

定する時は標準形式の仲介番号 (ARPANET ではソケット, NPL 網ではチャンネル⁽³⁾) を用いる。又、一度の接続手順で設定される仮想通信路はリンクなどと呼ばれているが、ARPANET では単方向性のリンクを用いており、その上を流れるメッセージの方向は定まっている。従って会話形通信を行う場合は、接続設定を2回必要とする。一方、NPL の提案や我が国の新データ網計画などでは、双方向性リンクになっており、メッセージは双方向に流れることができる。

NCP 間のメッセージである制御コマンドの送り方にも2とおりあり、始めから固定的に定められた制御リンク上で送る方法と、メッセージヘッダ内のタイプフィールドによって諸コマンドを送る方法とがある。前者の例は ARPANET, 後者の例は NPL 網である。

NCP を既存 OS の中に組込む方法⁽²⁷⁾ としては、NCP をスーパーバイザルーチンとして組む方法と、特権を持ったユーザプログラムとする方法が考えられる。一般に、既存 OS の修正はできる限り少なくしたいので、多くの場合ユーザプログラムとして組む方が容易である。しかし、今後、網が発達してゆくに従って、製作される OS が十分それを意識したものになってゆくことは十分考えられる。

3.3 ハイレベルプロトコル^{(3),(10)}

HOST/HOST プロトコルを基にしてハイレベルプロトコルが作り上げられる。網を利用したあらゆる種類の処理形態に対して適当な便査手段を設けることは当然必要であり、ユーザプロセスが他のプロセスと容易に通信しうるためにも是非必要である。しかし、一般にこのハイレベルプロトコルに関して、網上で全 OS に共通の取定めをまとめることはかなり難しい。

例えば、ファイルを転送するにはそのファイルのヘッディング内でそのファイル構造についての十分な記述をしておく必要があり、又、その記述を全 OS が同一に解釈できねばならない。端末についても同様で、図4のようにローカルな計算機を通して遠隔計算機を会話形利用したい場合、その端末の仕様が統一されている必要がある。従って、網上で標準の端末仕様を定

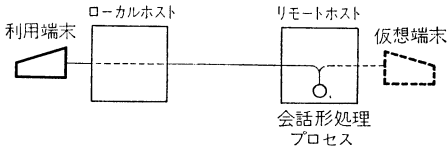


図4 網仮想端末

OS が異なるシステム間で任意のデータセットを伝送する網

③ 任意のシステム間で任意のデータセットを伝送する網

①には多くの例⁽¹²⁾があるが利用形態が RJE に限られており、データ記述メッセージは、ユーザ ID、ジョブネーム、そのジョブ内のデータセットを定義するネームくらいでよく、最も少ない。

②については、この種の網内で任意のデータセットを授受できるような通信インタフェースの研究がある。これは、単なる RJE に止まらずより汎用化された網利用をサポートするため、IBM 研究所で設計・実装されたもの⁽¹²⁾である。CITASK (Communication Interface TASK) と呼ばれているが、この場合のデータ記述は送受両システムにおけるそのデータセットの記述が必要なので、例えば次のようなダブルリストの形を取る。

SND(...DSORG=PS,...),
RCV(...DSORG=VS,...)

これは、フィジカル順編成のデータセットをパーティシャル順編成のデータセットとしてファイルすることを示している。

③の1実験例は、ミシガンの3大学網 MERIT で見られる。異なるシステム間で任意のデータ転送を可能にするため、MICIS 標準のデータ記述言語⁽¹⁴⁾が提案されている。このデータ記述は網内で同様にエンコードされる標準文字セットで書かれ、その最初にはデータ記述自身のレイアウトや配置を定義するブートストラップレコードが付く。データは配列として幾つかのグループに分けられ、各変数のレベルまでの記述がされる。しかし、単一のデータディスクリプタセットですべての応用を満たすことは非常に困難であり、MICIS 自体も統計のアプリケーションのみを目的としたものである。

一方、データの貯え方を統一し、それによってデータ変換の問題を避けようという試みもなされている。すなわち、データをすべて直列の本のような形式で保存し、章・節・文など、普通の本の言葉を用いて参照する方法⁽²⁹⁾である。

4.2 データ記述言語とアクセス法のモデル

データ構造の記述や、広範なデータベースアクセス用のモデルを開発する試み^{(15),(16)}は幾つかなされている。

例えば IBM の Data Independent Architecture

め、各計算機システム内でローカルな端末とのマッチングを取る必要がある他、端末 ID を計算機に通知してインタラクトを開始するための標準手法が必要である。

このようなプロトコールには次のようなものが考えられている。

- ① 網仮想端末プロトコール
- ② リモートジョブエントリプロトコール
- ③ データ転送プロトコール
- ④ ファイル転送プロトコール
- ⑤ メッセージメールボックスプロトコール
- ⑥ グラフィックスプロトコール

4. データ記述^{(11)~(20)}

4.1 網におけるデータ記述⁽¹²⁾

最近の計算機システムでは、ユーザプログラムはその利用データの存在場所・レイアウト・処理手法などのデータ記述を行うのが普通であるが、計算機網では利用したいシステム上で動かすすべてのプログラムに対して、データ記述が特に必要になる。それは計算機間でのデータ表現の差、ファイルコンベンションの差が問題になるからである。

これに対する対策としては、データの供給側が多くソフトウェアインタフェースを設けるか、ユーザがそれを作るか、もしくはハイレベルのデータ表現プロトコールを設けデータのユーザとサーバがそれに従うかであろう。この内最後のものは、各データセットに対しそれを別のシステムの外部記憶に置く時に必要な詳しい記述(個々の変数や、そのビットストリングの分解に至るレベルまでの)を付けておくもので、付加されるデータ記述は網内のどこでも解釈できるような形でなされておらねばならず、標準のデータ記述言語を想定した方法である。

データ記述の難易度は網によって異なる。容易さの順に網を分類すれば次のようになる。

- ① 両立性のある OS と計算機を要素とした RJE 網
- ② データの内部表現は同一だが、ハードウェアや

Model は4階層から成る。最上層は、データエンティティとそれらの間の関係によってデータを見る抽象的なデータ記述で、その下にはアクセスパス構造を記述するレベルがあり、更にその下にアクセスパスの各要素を記述するレベルがある。そして最下層には物理デバイス上のエンコーディングモデルがある。

又、蓄積データ定義言語 (SDDL) としては、CODASYL のタスクグループが作ったもの⁽¹⁶⁾がある。これは2つの独立したデータ記述から成る。1つは蓄積データの論理的な面を記述し、他はそのデータが置かれている蓄積媒体の構造を記述するものである。

その他、両立性のないシステム間でのファイルの読み書きを可能にすることを目的とした、汎用アクセス法モデルの研究⁽¹⁵⁾がある。これは、デバイス記述やデータアクセス法を高級言語によって記述しようとするものである。

4.3 データ変換および蓄積サービス

計算機網で利用されているデータ変換サービスや情報検索サービスとして、ARPA 網の Data Reconfiguration Service と、MDC-Programmer とについて述べる。

(1) データ再編成サービス (DRS)^{(17), (18)}

DRS はデータ変換に対する1つのアプローチとして実施されているものである。DRS では、アプリケーションプログラマが、必要なデータ再編成を記述するフォームと呼ばれるプログラムを書く。DRS はこのフォームをそのネームと共に蓄積し、後ほどその利用者が必要な時に DRS を用い、ユーザプロセスとサーバプロセスの間のデータストリームにその変換を施す。この時、利用者はそのフォームをネームで呼び出し、データユーザプロセスとサーバプロセスとを指定

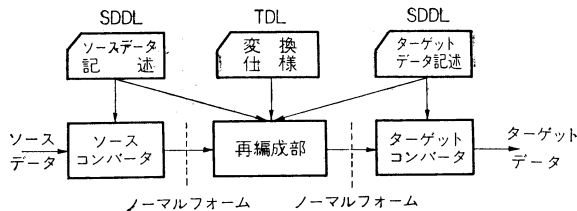


図6 データトランスレータモデル

しさえすればよい。現在 ARPANET では、UCSB と MIT にそのインタプリタがある。

汎用のデータ変換プロジェクトとしては、他にミシガン大学のデータトランスレータモデル⁽¹³⁾がある。これはトランスレーション定義言語 (TDL) と、蓄積データ定義言語 (SDDL) とから成り、図6のように一度標準形式のデータに直してから変換を行うものである。

(2) MDC-Programmer⁽²⁰⁾

これは大容量 (10^2 ビット) のオンライン記憶を持ったデータコンピュータの情報蓄積・検索サービスと、強力なプログラミング言語である Muddle とを結合したサービスで、データコンピュータが持っていないデータ処理機能を付け加えることにより、非常に有用かつ強力なサービスになることが期待されている。データコンピュータは、Computer Corporation of America によってオペレートされている ARPANET 上のサービスであり、高度な構造を持った情報管理に特に向いている。一方、Muddle は MIT で設計されたもので、LISP のより発展したプログラミング言語である。LISP よりも利用できるデータタイプやシンタックスが多く、拡張性に豊み、計算機網に対するインタフェースを持っている。

5. 網ユーザーサービス^{(11), (21)~(26)}

計算機網は、リソースが多く又その多様性に豊んでいる程一般に有用性が増すと考えられるが、逆にユーザから見た場合⁽²¹⁾、特別な援助システムが無いと利用が増々難しくなる。特に ARPANET のような多目的・異機種種の網ではそうである。この対策としては、リソース情報を得る手段を改善すること、リソースへのアクセス手続きの多様性に対処すること、特定の HOST を持たない端末ユーザからの網への参加を容易にすることなどである。

5.1 オンライン情報サービス

網におけるオンライン情報サービスは機能上次の³

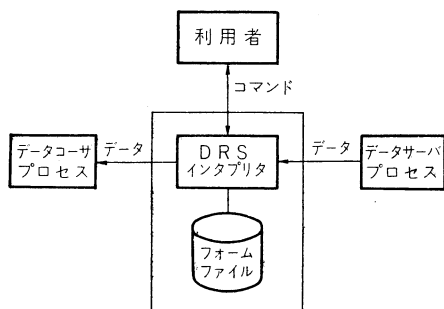


図5 データ再編成サービス

グループに分けられる。

① ヘルプファシリティ

会話形処理サービスを提供する単独システムにおいて現在、普通に見られるもので、例えばコマンドプロンプティング、用語チェック、オンラインドキュメンテーションなどがある。

② ユーザサービスディレトリ

網内各計算機システム上で利用可能なリソースについての情報を集中蓄積したもので、この良い例は ARPANET の網情報センタのリソースノートブックであり、オンライン情報サービスや出版印刷も行っている。

③ リソース管理サービス

②ではリソース相互の関係情報は得られないし、リソース取得も不可能であるが、③はリソースについての比較情報を提供し、リソースの選択・取得の援助をする。このサービスの最初は BBN 社で ARPANET 上のシステムとして開発された Resource Sharing Executive⁽²³⁾ であり、例えば、次のようなサービスが提供される。

- (i) TENEX ホストにログイン中のユーザ位置を知る。
- (ii) 端末を他の TENEX ユーザ端末とリンクする。
- (iii) 幾つかの TENEX ファイルシステムを結合して1つの複合ファイルディレクトリを作成すると同時にそのファイル操作を可能にする。
- (iv) 特定の HOST を持たない端末 (TIP) ユーザに対し、ログイン前の状態で網のステータスやニュース、及び幾つかの機能を提供する。

この他、③に属するものとして MITRE Co. の開発している REX システム⁽²²⁾がある。これは、ARPANET 上で開発されている網リソースマネージャのプロトタイプであり、端末ユーザが簡単な会話形問合せ言語を用いることによって特定のシステムに対するソース情報を提供する他、逆に特定のリソースの位置やその情報をも提供する。更にそのリソースを取得する方法をも提供する。最終的な目標は、REXに、ユーザが選んだリソースと結合させる諸プロトコルを実行させることであるが、現在のバージョンでは実際のリソース取得は幾つかのチュートリアルヘルプサービスとの結合のみに限定されている。例えば、コマンド

FIND FORTRAN

はその結果として、FORTRAN を持つ HOST のリ

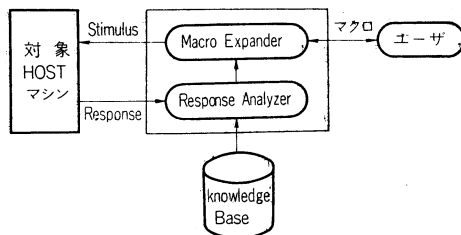


図7 ネットワークアクセスマシン

ストが得られるし、ACQUIRE というコマンドでチュートリアルサービスとの結合が行われる。

5.2 ネットワークアクセスマシン

異種計算機網で様々なリソースを利用するには、ユーザは多くの異なったリソースアクセス手順を憶えねばならない。例えば、初期接続、ログオン、ログオフ、切断などの手続きがそうである。

この対処法として、ユーザと網との間にミニコンピュータを置き、各リソースへの適当なアクセス手続きを自動的に生成することが考えられる。これは Network Access Machine (NAM) と呼ばれ、ARPANETのために NBS で開発されている^{(24),(25)}。

NAM は、プログラムジョブ制御の機能をユーザに提供するもので、網内の各計算機システムの評価道具や、コマンド標準化への試みに必要な道具としても利用できるものである。

NAM は図7のような形を取る。NAM は諸リソースへのアクセス手順を収めたファイルディレクトリを保持しており、その参照を行うことがユーザの個人的な共通アクセス言語の基礎になっている。このアクセス手順はユーザが定めるが、マクロの形をしており、それを NAM が展開して所要リソースとの固有の対話を生成する。この対話は、そのリモートシステムへ送られるメッセージ (Stimulus と呼ぶ) と、そこから返される応答 (Response) である。

5.3 計算負荷の自動バランシング

網内の計算機各々にかかる負荷のバランスを取り、全体としてジョブのターンアラウンド時間を短く、全リソースの有効利用をはかるため、計算負荷の制御と移動を行うことが考えられている^{(26),(27)}。負荷の移動が可能であるためには、相互の計算機に両立性があることと、更にそのジョブがオンライン記憶のみにアクセスし、磁気テープや生のデータを利用しないものでなければならない。負荷の制御を行うためには、各計算機にかかっている負荷を測定する必要があり、例えば現在の待ちジョブ数の利用が考えられる。又ジョブ

の移動決定には、そのジョブの推定処理時間、転送や結果受信に要する時間とそのコストなどを総合的に考慮して決めなければならない。

負荷の制御と移動システムのモデルとしては、どこかに網負荷測定と制御のセンタを置く方式⁽²⁶⁾が検討されている他、比較的近距离の網では、計算リソース要求に対して各プロセッサが入札を行い、その内最もコストが安いものを自分(要求を出したプロセッサのOS)が選ぶという分散リソース管理システム(Distributed Computer System)⁽²⁸⁾も考えられている。

6. おわりに

計算機網における主要な問題は、それを有効に利用するための諸ソフトウェアである。本文ではその内、特に基本制御システムとデータ記述問題、および網サービスを取りあげた。

網が大きく成長するにつれて、その基本制御システムの改良は増々困難になってくる。これは標準化のほう1つの側面であるが、計算機網は一部において既にそのような段階にまで、達しているようである。

文 献

- (1) 田中英彦：“コンピュータ・ネットワーク”，信学誌，ネットワーク特集号，58，4，p. 338(昭50-04)。
- (2) 高月敏晴：“広域電子計算機網 [I]，[II]，[III]”，信学誌，57，5~7 (昭49-05~07)。
- (3) D.W. Davies and D.L.A. Barber：“Communication networks for computers”，John Wiley & Sons Inc. (1973)。
- (4) Special Issue on Computer Communications, IEEE Trans., COM-20, 3 (June 1972)。
- (5) “計算機システム技術の最近の動向—新技術動向調査報告”，日本電子工業振興協会，49-C-277 (昭49-03)。
- (6) E.W. Dijkstra：“Co-operating sequential processes”，Programming Languages, Academic Press, New York (1968)。
- (7) D.C. Walden：“A system for interprocess communication in a resource sharing computer network”，Comm. ACM, 15, 4 (April 1972)。
- (8) A. McKenzie：“HOST/HOST protocol for the ARPA network”，NIC-8246 (Jan. 1972)。
- (9) D.B. McKay and D.P. Karp：“Protocol for a computer network”，IBM Syst. J., 12, 1 (1973)。
- (10) S.D. Crocker, et al.：“Function oriented protocol for the ARPA computer network”，SJCC 1972, AFIPS, p. 271。
- (11) “計算機アーキテクチャの最近の動向—新技術動向調査報告”，日本電子工業振興協会，50-C-292 (昭50-03)。
- (12) D.H. Fredrickson：“Describing data in computer network”，IBM Syst. J., 12, 3, p. 257 (1973)。
- (13) J.P. Fry, R.L. Frank and E.A. Hershey, III:

- “A developmental model for data translation”，Proc. ACM SIGFIDET Conference on Data Description, Access and Control (1972)。
- (14) M. Donaldson, S. Robinovitz and B. Wolfe：“Preliminary draft: proposed MICIS standard for data description”，Michigan Interuniversity Committee on Information Systems (Dec. 1970)。
- (15) R.L. Frank and K. Yamaguchi：“A model for a generalized data access method”，NCC 1974, p. 45。
- (16) Stored Data Definition and Translation Task Group：“An approach to stored data definition and translation”，Proc. ACM SIGFIDET Conference on Data Description, Access and Control (Nov. 1972)。
- (17) V. Cerf, E.F. Harslem, et al.：“An experimental service for adaptable data reconfiguration”，IEEE Trans., COM-20, 3, p. 557 (June 1972)。
- (18) E.F. Harslem, J. Hearfner and T.D. Wisniewski：“Data reconfiguration service compiler: communications among heterogeneous computer centers using remote resource sharing”，ARPA Report, R-887-ARPA (April 1972)。
- (19) T. Marill：“Network data handling system”，CCA Semiannual Technical Report, AD-730724 (Sept. 1971)。
- (20) S.A. Bengelloun：“MDC-programmer: a muddle-to-datalanguage translator for information retrieval”，Technical Memorandum, Project MAC, MIT (June 1974)。
- (21) J.R. Pickens：“Computer networks from the users point of view”，COMPCON 1973, preprint p. 71。
- (22) J.W. Benoit and E.G. Webster：“Evolution of network user service—the network resource manager”，Proc. 1974 Symposium on COMPUTER NETWORKS: Trend and Applications, p. 21。
- (23) R.H. Thomas：“Resource sharing executives for the ARPANET”，NCC 1973, AFIPS p. 155。
- (24) R. Rosenthal and S.W. Watkins：“Automated access to network resources, a network access machine”，Proc. 1974 Symposium on COMPUTER NETWORKS: Trend and Applications, p. 47。
- (25) 弓場敏嗣：“Network Access Machine の設計”，電総研費報，38，11，p. 654 (昭49-11)。
- (26) J.T. Fitzgerald：“Load regulation and dispatching in a network of computers”，Univ. of Illinois, UCDCS-R-72-537 (Aug. 1972)。
- (27) R.H. Howell：“The integrated computer network system”，ICCC 1972, p. 214。
- (28) D.J. Farber：“The structure of a distributed computing system—software”，Symp. on Computer-Comm. Networks and Teletraffic (April 1972)。
- (29) B.T. Smith：“Mixed computer networks: benefits, problems, and guidelines”，ICCC 1972, p. 201。

- (30) 田中英彦：“ARPA ネットワーク”，計測自動制御学会誌，14，8，p. 579（昭50-08）。
- 計算機間通信（データ通信）については，上記(1)，(2)，(3)，内に示されている文献または(30)を参照されたい。