

解説

コンピュータ・ネットワークにおける通信制御用コンピュータ*

石田 晴 久**

1. リモート・バッチ網からコンピュータ・ネットワークへ

最近わが国でも関心もたれているコンピュータ・ネットワーク¹⁾は、1組の大型機を中心に星状に広がる従来のリモート・バッチ網、TSS網、オンライン網などのさらに高次の拡張とみることができる。このコンピュータ・ネットワークが従来のデータ通信網と異なる面としては、次のような点があげられる。

(1) ネットワークでは、ホスト・コンピュータ同士が対等の資格で結ばれる。これに対してリモート・バッチ網などでは、中央コンピュータと端末コンピュータとの間に明白な親子関係がある。

(2) ネットワークでは、1本の回線(物理リンク)の上でメッセージの多重化が行われ、複数個のジョブ(プロセス)のメッセージが同時に(ただし時分割で)1回線上で送受できる。リモート・バッチでもこれは可能であるが実際にはほとんど行われていない。

(3) 将来サブネット(通信網)として新データ網が使えるようになれば、回線はオール・デジタルで高速となり、メッセージ交換が行われて、通信料金は従量制となる。これは現在のデータ通信がほとんど専用線で行われ、交換は電話回線網(公衆網)利用のときしか行われていないのと対照的である。

以上のようなコンピュータ・ネットワークの特長から整理してみると、コンピュータ・ネットワーク内で通信制御を受持つコンピュータは図-1のように分けられるであろう。図-1のFEP(Front End Processor)はなくてその機能がホストあるいはIMPに含まれることもあり、IMP(Interface Message Processor, 中継用ミニコン)²⁾には、TSS用端末機が直接つなげる

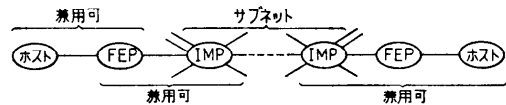


図-1 コンピュータ・ネットワークにおける通信制御用コンピュータ

TIP (Terminal IMP)³⁾ やリモート・バッチ端末である NTS (Network Terminal System) もある。またインハウス・ネットワークではホストが FEP のみならず IMP を兼ねることもある。1台のホストに何台かの FEP あるいは IMP をつなげることももちろん可能である。

さて、通信制御用コンピュータとしての FEP と IMP との主な違いは、IMP ではメッセージ交換(スイッチング)が行われるのに対し、ホストの単なる下請けである FEP はそれを行わない点にあるから、FEP は IMP の1部とみてよい。

一般に FEP を含む IMP のもつべき機能には以下のようなものがある。

- (a) ホストおよび通信回線側との交信
- (b) メッセージの蓄積交換
- (c) メッセージの経路選択(ルーティング)
- (d) メッセージのフロー制御
- (e) メッセージのシーケンシング(順序制御)
- (f) 障害回復
- (g) ネットワークの監視および動作の測定

通信制御用コンピュータの規模としては通常ミニコンで十分である。ただメッセージのバッファはなるべく多くとれる方がよいから、メモリは多いほどよい。周辺機器としてはホストおよび回線とのインタフェースが必須なほかに、テレタイプおよびプログラム・ロードのための高速紙テープ読取機が最低限あればよい。コスト的にはミニコン本体よりも、ホストおよび回線とのインタフェースが全く標準化されていないので高くつく。インタフェースの標準化は早急に望まれるところである。

* Communication control computers in computer networks by Haruhisa ISHIDA (Computer Centre, The University of Tokyo)

** 東京大学大型計算機センター

次に通信制御用コンピュータとして非常に面白いのは ARPA ネットワーク⁴⁾で、1975 年から使用される予定のマルチ・ミニ IMP である⁵⁾。これは Lockheed SUE と呼ばれる 4 kW メモリ付きの低コスト低速ミニコンを 32 kW の共通メモリに 14 台つないだ構成になっている。したがってメモリは 1 語 16 ビットで全体で 88 kW になるが、各 CPU 固有のメモリには使用度の高い同じプログラムを入れておき、共通メモリはバッファや共通変数領域として使われる。全体として全 2 重で 750 キロビット/秒のスループットを持ち、20 台以上のホスト、数百台の端末をサポートすることを目標としているという。また CPU はどれかがダウンしても全体としては動くようにし、信頼性を高めることを目標としている。

2. 通信制御のためのインタフェース

次に FEP あるいは IMP を設計・選択するときにはまず問題になるのは、一方でホストとのインタフェース、他方で回線(またはサブネット)とのインタフェースである。インタフェースには、ハードウェア上のものと、ホスト IMP (または FEP) プロトコルあるいは IMP-IMP プロトコルと呼ばれるソフトウェア上のものがある。いずれも必要に応じてどちらかを単純にし、他方にその分の負担をおわせることが可能である。

まず対ホストのハードウェア・インタフェースを考えよう。図-2 は簡単なものの例で、ARPA ネットワークでのインタフェースである。ここでの転送はビッ

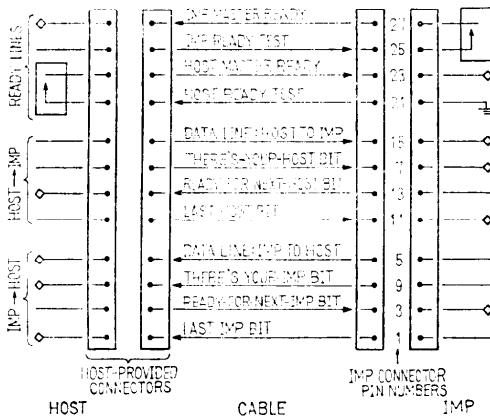


図-2 ARPA ネットのホスト IMP インタフェース

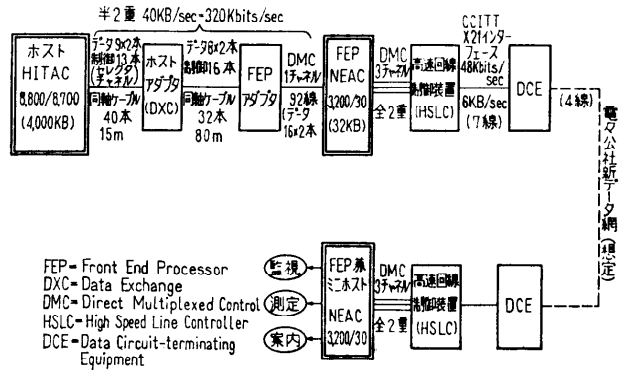


図-3 コンピュータ・ネットワーク・システムの接続例

ト直列に行われるが、直並列変換はハードウェアによるため、転送速度が 50 kbits/sec の場合でもバイト当りの時間は 160 μsec となるからミニコンの速度でも十分に追従できる。一方図-3 は複雑なインタフェースの例で、東大の実験システムで使われている構成を示す⁶⁾。これは JIPNET⁷⁾ のインタフェースを一部改造したもので、新しいホストの接続に便利のように、ホストに依存しない FEP 共通アダプタとホストごとに作るホスト個別アダプタに分けられており、両側ともチャンネル接続になっている。ここでデータはホストと FEP アダプタ間では 8 ビット並列転送であるが、FEP とそのアダプタ間では 16 ビット並列となっている。スピードは 1 メガ bits/sec 程度まで可能であるが、ホストに余り負荷をかけないようにするため 320 k bits/sec におさえ、しかもホストとの間は半 2 重通信にしている。

図-3 のホスト FEP 間のプロトコルは、片方が Write Data コマンドを出し、他方が Read Data を出したときに転送が行われることを基本にしている。相手が Read Data コマンドを出していない時は割り込みをかける機能もある。またアダプタ内には双方がデータを転送できるかどうかを示すプログラム・スイッチが一つずつ設けてあって、Lock, Unlock コマンドでセット、リセットができるようになっており、それらのセンス・コマンドも用意されている。

一方回線側のインタフェースの方は、普通は通信制御装置を通してその先はモデム(変復調装置)インタフェースとなる。しかし、伝送制御手順に今後の国際的な標準となると考えられる高水準データ・リンク制御(HDLC, IBM では SDLC)⁸⁾ 手順を採用し、サブネットとしてデジタルの新データ網を想定すると、

図-3の右半分のように、不要となるモデムの代りにDCEに対するインタフェースが必要となる。このDCEには、相手ホストをパルス列の形の番号で呼び出すためのキャラクタ・ダイヤルの機能がつくのが新データ網の特徴である。なお図-3の東大の実験システムでは、HSLCの接続には、送信データ、受信データ、割込み信号にそれぞれチャンネルを1本ずつ使う複雑な全2重インタフェースを使っているが、FEPの能力にゆとりがあれば、これはもちろんもっと簡単にしてもよい。

以上のべたHSLCのような通信制御装置やホスト・インタフェースは今のところ布線論理回路で組まれているが、最近の傾向としてこうした回路にマイクロコンピュータを採用する例が増えつつある。たとえばベル研究所のSPIDER⁹⁾と呼ばれるリング状(データ・ハイウェイ方式)のネットワークには10台程度のホスト(主にミニコン)がつながれているが、回線リングと各ホストとの接続部に使われているのはマイクロコンピュータである。最近は各所で高水準データ・リンク制御(HDLC)にマイクロコンピュータを使うことも試みられている。

3. 通信制御上の問題

次に通信制御用コンピュータのソフトウェアを考える前提として、ネットワークの通信制御で考慮しなければならない問題¹⁰⁾をあげてみる。

(1) メッセージの蓄積交換

ホスト、FEP、サブネット(IMP)の間でメッセージのやりとりをする時に問題となるのは、各コンピュータの中でのバッファの長さをどうとるかである。もちろんメモリの効率的利用のためには、バッファは可変長の方がよいが、それではソフトウェアの効率が落ちるし、チャンネル利用にも不便なので普通にはバッファは固定長にする。その際問題はバッファの長さをどの位にするか、バッファを何個用意するか、である。例をあげれば次のようになる。

バッファ長(語=16ビット)			4kW中にとれる バッファ個数
ビット長	バイト長	語数	
8,192b	1,024B	512W	8個
1,024b	128B	64W	64個

通常タイプライタでの1行、カード1枚、ラインプリンタでの1行などの含む情報は100バイト前後だから、1,024ビットの長さは大体それに相当する。これ

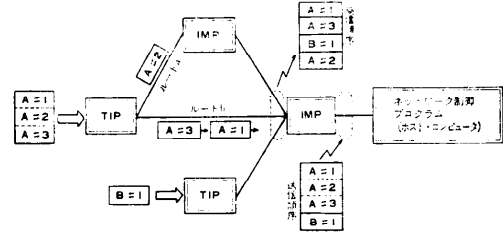


図-4 パケットの伝送とルーティング

に対しホストで扱うメッセージは一般に不定長であるが、それをホスト内で8,192ビット程度の“セグメント”に区切ることもある。それをさらに1,024ビット程度の“パケット”に区切るのをホスト側でやるかFEPまたはIMPでやるかも決めなければならない問題である。メッセージをパケットに分割することをデイスアSEMBリ、逆にパケットをメッセージに組立てることをリアSEMBリと呼ぶ。メッセージ交換に関しては、図-4にも示したように、ホスト側のIMPでは各メッセージごとにバッファを用意し、バラバラに順序不同で送られてくるパケットを振分けてリアSEMBルする機能がある。

(2) ルーティング(経路選択)

これはARPAネットのように中継点(ノード)の多い網内のIMPでの問題で、図-4のように、あるIMPから相手ホストへ至る経路がいくつもあるときに、どのルートを選ぶのがよいかということである。特定IMP間の回線ダウンの検出もこれに含まれる。ルーティングにはいろいろなアルゴリズムが提案されているが、現実にはルーティングが問題になるほど大規模なネットワークは余りないであろう。将来の新データ網ではルーティングは網側の管理となる。

(3) フロー制御

メッセージの流れ(フロー)の制御は、ホストやFEPやIMPのバッファが有限個しかないために、メッセージ(パケット)がうまく送れなくなるという状況避けるためにネットワーク内で必要となる機能である。これはとくにルートのたくさんあるネットワークでは、リアSEMBリ・ロックアップや蓄積交換(store and forward)ロックアップと呼ばれるデッドロック現象を防止するのに必要となる。

図-5(次頁参照)にリアSEMBリ・ロックアップの状態を示す。図ではメッセージAとメッセージBがそれぞれ三つのパケットに分けられて別々のルートで送信され、たまたまA1とB2というパケットの到着が遅れている状態を示す。この場合、IMP3はパケットの

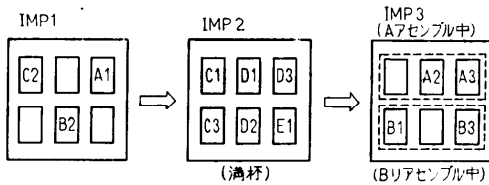


図-5 リアセンプリ・ロックアップ

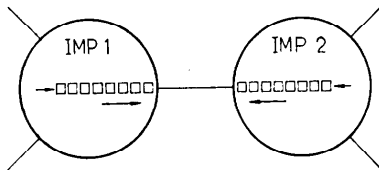


図-6 蓄積交換ロックアップ

順序を揃えてメッセージの形に組立ててからそれをホストに渡すものとする、IMP3の空きバッファはA1とB2のためにとっておかなければならない。そうするとIMP3に本当に空いているバッファがないため、IMP2からIMP3への送信もできず、IMP1からの送信もできず、パケットは全く動けなくなる。

一方図-6は蓄積交換ロックアップの状態を示す。これはIMPのバッファがたまたますべて送信用パケットで満杯になった状態である。こうなると相手に空きバッファがないためお互にパケットを送出しようにもパケットは全く動きがとれなくなる。しかしこの状態は受信にも必ずバッファを割当てることにしておけば防止できる。したがってフロー制御で問題になるのはリアセンプリ・ロックアップである。

リアセンプリ・ロックアップ防止策として通常とられるのは次のようなARPA方式である。

- (a) ホストのプロセス間で、物理的な回線とは独立に、プロセスとプロセスをつなぐ論理的な一方向性回線(リンク)を考え、リンクごとにある数のバッファを確保する。
- (b) ひとつのリンク上では一時には1個のメッセージしか流さない。これにはあるメッセージ(すなわちそのメッセージに属するすべてのパケット)が相手ホストに届いたことを送信ホストが確認してからでなければ次のメッセージを送らないことにすればよい。ARPAネットでは相手ホストからの確認メッセージをRFNM(Ready For Next Message, レフナム)と呼んでいる。
- (c) 送出の際相手に空きバッファがあることをまず確認する。ARPAネットのように、メッ

ージの最大長が限定されていて、それが最大8個のパケットに分割されるときには、リンクごとに確保すべきバッファの数は最大8個でよいことになる。ただしこの場合、相手IMP側で本当に8個までのバッファを確保してもらうために、送信IMPから受信IMP側に「 n 個のバッファを確保してくれ」(RFA=Request For Allocation)という指令を出し、受信側から「確保した」(Allocate)という返答をもらう必要がある。しかし実際には数からいうと多い1パケット長のメッセージに備えて、要求がなくてもリンクごとに少くともバッファ1個は用意しておくことが多い。

なおフロー制御にはバッファ制御の1方式であるウィンドウ方式や、ネットワーク内のトラフィック量の総量規制を行って網内の混雑を一定限度内におさえるためのIsarithmic(equal numberの意)方式すなわち切符(permit)発行方式も提案されている¹¹⁾。このうちウィンドウ方式はひとつのメッセージを端から一定の幅(windowと呼ぶ)だけとって送り、次にそのウィンドウをずらして次々に一定幅ずつを送る方式である¹²⁾。これはひとつのリンクを通していくつかのメッセージを同時に送るメッセージ多重化の目的にも使われる。

(4) シーケンス制御

メッセージの多重化には、以上のべたリンクーウィンドウ方式の他にも、リンクの代りに4個までのメッセージが同時に流せるパイプを2種考える方式(ARPA)、さらにパイプに3種の区別を考える方式(JIP-NET)もある。これらの場合、1本のパイプを通して4個のメッセージに属するパケットがバラバラに送られてくることになるから、それらを各メッセージごとに正しい順序に並べる必要がある。これがシーケンス制御である。

(5) 障害対策

ネットワークでは、その1番低レベルのIMP-IMPプロトコルに相当する伝送制御手順としてHDLC手順を使う場合には、サイクリックな冗長度検査にもとづくエラー検出が行われ、必要に応じてパケットの再送が自動的に行われるから、パケットが紛失する確率は非常に小さいと考えられる。しかし万一に備えて、パケットを送出した側で、一定時間内に相手から受信確認通知が戻ってこないときは、自動的に再送するか、あるいは受信側で一定時間たっても後続パケット

が到着しない場合に再送を要求するなどの時間監視による障害管理を行う必要がある。

またサブネット内の IMP は絶えず回線の接続状況を監視していなければならない。これには応答が全く返ってこない回線を検出するのによいし、専用線利用の場合なら、相互に送るべき情報がないときには、片方から Hello メッセージを送り、相手から I heard you というメッセージが返るかどうかをみる手もある。さらにホストや IMP がダウンしたらそれを早目に検出する機能も必要である。ARPA ネットではこうした障害情報を集めて通報するためのコンピュータをネットワーク管制センターに置いている。

(6) バックグラウンド処理

これには統計情報やルーティング情報の収集と報告が含まれる。ARPA ネットの IMP では統計情報領域をメモリ内にとり、平均メッセージ(パケット)長のヒストグラム、メッセージ数、ACK の送信数・受信数、RFNM の送信数、受信語数・送信語数、パケット再送数、エラー数などを収集している。

5. 通信制御用コンピュータのソフトウェア

FEP や IMP のソフトウェア (TCP=Transmission Control Program) は、普通簡単なモニタを置き、通信制御用の各種タスクが割込みによって起動されるような形に構成される。たとえば JIPNET を例にとると、IMP ソフトウェアの構成は表-1 のようになる。これらのタスクの起動のされ方は IMP のはたらきを考えると分かり易い。まずメッセージが短くて1個のパケットに収まる場合、あるいは長いメッセージでも入る位のバッファが用意できる場合を次にあげてみよう。ただしルーティングは省略する。

(1) ホスト・インタフェースを通して、ホストからメッセージを READ してバッファに入れる。

(2) バッファを読み出し相手 IMP へ送信する。

表-1 通信制御プログラムの例

モニタ (割込み処理) (優先レベル3)	
— 回線受信タスク (レベル3)	
— 回線送信タスク (レベル1)	
— ホスト・インタフェース・タスク (レベル1)	
— タイマ・タスク (レベル1)	
— 他の割込み処理タスク (レベル1)	
— タスク・スケジューラ (レベル0)	
	— バイア (リンク) 管理 — シーケンス制御
	— ルーティング
	— 制御パケット処理 (RFA, ALLOCなど)
	— ディスアセンブリ (分解)
	— リアセンブリ (組立て)
	— バックグラウンド処理

バッファ内容は再送に備え消さないでおく。

(3) 相手 IMP より受信確認通知 (RFNM) を受信したら、バッファの内容を消す。

(4) 相手 IMP よりメッセージを受信したらバッファへ入れる。

(5) 回線へ向けて ACK を返す (送信する)。

(6) バッファの内容をホストに向けて WRITE する。

以上のようにメッセージすなわちパケットの単パケット・メッセージの場合は通信制御は比較的簡単である。ところがメッセージが複数個のパケットからなるマルチ・パケット・メッセージになると ARPA 方式では次のように通信制御はかなり複雑になる。

(1) ホストよりメッセージを READ してメッセージ・バッファ (たとえば8,192ビット=512語) へ。

(2) バッファ中のメッセージをパケットに分解し、メッセージ番号やパケット番号などを含むヘッダをつけながら、パケット・バッファ (たとえば1,024ビット=64語) に順々に入れる。相手 IMP とリンク番号はメッセージの先頭部分 (リーダー) から分るので、それをもとに送信リンク・テーブルを作る。

(3) 相手 IMP に RFA パケットを送信して、パケット個数分のバッファを要求する。

(4) 回線より RFA に対する ACK を受信する。(ACK に対する ACK は返さない。相手 IMP からやがて RFNM が戻るが、それに対する ACK も返さない)。

(5) 相手 IMP より Allocate パケットを受信し、相手側に確保されたバッファ数を知り、RFNM を返す。

(6) 回線に対して Allocate に対する ACK を返す。

(7) 相手 IMP のバッファ数分だけパケットを次々に送信する。パケット・バッファは消さない。

(8) 回線より各パケットに対する ACK を受信する。さらに未伝送のパケットが残っていれば(3)から繰返す。

(9) パケットの伝送がすんで RFNM が来たらバッファの内容を消す。

(10) 相手 IMP から受信したパケットが RFA (バッファ要求) であれば (RFNM を返し)、バッファを割当てる。(空きバッファがないときは、このリクエストをテーブル (キュー) に登録する。)

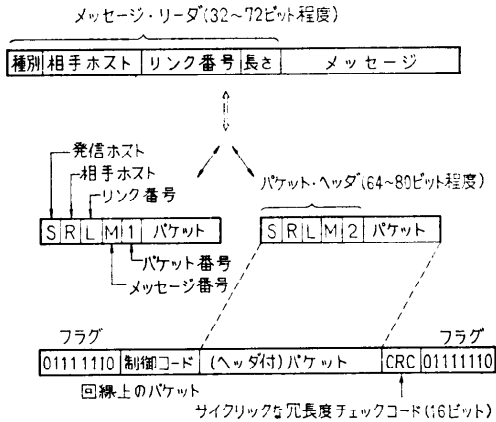


図-7 メッセージとパケット

(11) 相手 IMP に対し、割当てたバッファ数を用いて Allocate パケットを送出して知らせる。

(12) 回線より Allocate に対する ACK を受信する。(やがて RFNM が戻る。)

(13) 相手 IMP から次々にパケットを受信してそのパケット番号に対応するバッファに入れ、受信リンク・テーブルにその番号や時刻を記入する。

(14) 受信したパケットに対し ACK を返す。

(15) ひとつのメッセージに属するすべてのパケットを受信したら、それらのヘッダを取除いて、リーダーを付加しメッセージに組立て、メッセージ・バッファに入れて、RFNM を返す。

(16) ホストに対してバッファ中のメッセージを WRITE する。

以上のような機能を果たすには、どのホストとどのホストをつなぐ何番目のリンク上のどのメッセージのどのパケットかを示す情報が、図-7 のようにパケット・ヘッダとして各パケットについていなければならない。さらに回線上では、図-7 の 1 番下に示すように、たとえば HDLC 手順の場合、フラグや CRC などで 48 ビット程度のオーバーヘッド (レベル 0) を回線出力タスクで付けるから、パケットのヘッダによるオーバーヘッド (レベル 1) が 72 ビット程度とすると、オーバーヘッドは合せて 120 ビット、すなわち 15 バイト程度となる。平均メッセージ長を 30 バイトとすると、オーバーヘッドは全パケット長の 33% にもなるので、有効伝送速度やバッファ・サイズを考えるに当たってはこのオーバーヘッドは無視できない。

IMP では以上の機能のほかに、大規模なサブネットワークではルーティングや各種測定データの取得・報告を

行うことが要求される。各種の時間監視も必要である。たとえば、上記(4)(8)(11)でパケットを送出しても、ACK が返ってこないときにパケットを再送するのを待つ時間、あるいは上記(14)では、パケットが到着し始めてからそのメッセージに属するすべてのパケットが揃うまでの時間などが一定限度を越えたらタイムアウトにする必要が出てくる。

6. 今後の通信制御用コンピュータ

今後のコンピュータ・ネットワークの発達で期待したいのは、どんなところにあるどんなコンピュータでも、必要があれば割合に簡単なインタフェースと安い通信料金が使えるようになることである。この面からいえば、今後の通信制御用コンピュータに関して望まれるのは次のようなことであろう。

(1) 新データ網の発達によってサブネットに対するインタフェースおよび伝送制御手順が世界的に統一されること。

(2) ミニコンのメモリがさらに大きく安くなり、バッファ用に大きな領域がとれるようになること。

(3) 回線およびホストに対するインタフェース機能の大部分が LSI 技術の採用で低コストでハードウェア化されること。これは上の(2)と合せてソフトウェアの簡素化に大いに有効である。

(4) 通信制御用コンピュータの信頼性が大幅に向上し、その動作の測定・監視がやりやすくなること。

参考文献

- 1) 高月敏晴：広域電子計算機網 (I), (II), (III), 電子通信会誌, No. 5, 6, 7 (1974)
- 2) F. E. Heart, et al.: The interface message processor for ARPA computer network, Proc. of SJCC, Vol. 36, pp. 551~567 (1970)
- 3) S. M. Ornstein: The terminal IMP for the ARPA computer network, Proc. of SJCC, Vol. 40, pp. 243~254 (1972)
- 4) 安井裕：コンピュータ・ネットワークと大型計算機センタ, 大阪大学大型計算機センタ・ニュース, No. 15, pp. 1~18 (1975)
- 5) F. E. Heart, et al.: A new minicomputer/multi processor for the ARPA network, AFIPS Proc. of NCC, Vol. 42, pp. 529~537 (1973)
- 6) ネットワーク計画委員会：大型計算機センター間コンピュータネットワークの構成に関する研究, 科研費特定研究・広域大量情報の高次処理研究報告, pp. 276~298 (1975年3月)

- 7) 佐々木・桑原: コンピュータ・ネットワークの現状と問題点, 日経エレクトロニクス, pp. 49~91 (1974.6)
- 8) IBM のシステム・ネットワーク体系, 日経エレクトロニクス, pp. 66~79 (1974.12)
- 9) A.G. Fraser: A virtual channel network, Datamation, Vol. 21, No. 2, pp. 51~56 (1975)
- 10) H. Frank, et al.: Computer communication network design—experience with theory and practice, Proc. of SJCC, Vol. 40, pp. 255~270 (1972)
- 11) D. W. Davies: The control of congestion in packet-switching networks, IEEE Trans. on Comm., Vol. COM-20, No. 3, pp. 546~550 (1972)
- 12) V. Cerf and R.E. Kahn: A protocol for packet network intercommunication, IEEE Trans. on Comm, Vol. COM-22, No. 5, pp. 637~648 (1974)

(昭和50年4月15日受付)